CONTRIBUTIONS IN SUPPLY CHAIN RISK ASSESSMENT AND MITIGATION

A Thesis
Presented to
The Academic Faculty

by

Yu Zhang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
May 2013

CONTRIBUTIONS IN SUPPLY CHAIN RISK ASSESSMENT AND MITIGATION

Approved by:

Professor Alan L. Erera, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Joel Sokol
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Chelsea C. White III
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Lauren Davis
Department of Industrial and Systems Engineering
North Carolina A&T State University

Professor Jianjun (Jan) Shi
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 28 November 2012

To my family, teachers and true friends.

ACKNOWLEDGEMENTS

The time of pursuing knowledge and the doctoral degree is the most important chapter in my twenties. It has been my great privilege to study in the School of Industrial and Systems Engineering (ISYE) at Georgia Institute of Technology. The great experience that I have gained here will always be a treasure of my entire life.

First and foremost, I would like to express my heartfelt gratitude to my advisor, Dr. Alan Erera. He has patiently provided the vision, encouragement and advice necessary for me to go through the doctoral program and complete this dissertation. He has been a strong and supportive advisor throughout my graduate school career, while he has always given me great freedom to pursue my own ideas.

Special thanks to my committee, Dr. Chip White, Dr. Jan Shi, Dr. Joel Sokol, and Dr. Lauren Davis for their support, guidance and helpful suggestions. Their guidance has served me well and I am honored to have them on my committee.

The fellows in ISYE also deserve my sincerest thanks. Of special note, I would like to thank my colleagues Yang Zhang, Kate Lindsey and Yanling Chang for their assistance and support to my research. Moreover, I always appreciate the friendship of Feng Qiu, who has provided unselfish help to my study and life in Atlanta.

Last but not least, I would like to express my special thanks to my parents, Yufeng Zhang and Aizhen Liu, and other members of my large family. Their unreserved love makes me feel a sense of responsibility. In addition, I would like to extend my gratitude to all my teachers, who has taught me knowledge, and more importantly, the way of acquiring knowledge. Finally, I will never forget my true friends, who always love and support me, even though we do not often meet each other.

To all of them I dedicate this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation develops contributions in the area of supply chain risk assessment and mitigation. In each of the three main chapters, we present and analyze a risk assessment or mitigation problem for supply chains. The first problem is to assess the impact of infrastructure disruptions on supply chain performance; the second problem is to develop an operational control approach to mitigate risks posed by uncertain events that disrupt network synchronization; and the third problem is to analyze the risk posed by an adversary seeking to use a supply chain as a weapon.

Chapter II presents a methodology for assessing the excess supply chain costs that arise from a failure of or an attack on a critical supply chain infrastructure component. Different from many subjective risk assessment practices, our methodology provides a systematic approach to search for the most vulnerable supply chain components and measure the economic consequences of disruption. Modeling a supply chain using network flow models, we analyze the impact of disruption by linear programming theory, and propose an efficient assessment algorithm based on the dual network simplex method. Finally, a case study on the U.S. corn export supply chain is presented.

Chapter III discusses the mitigation of risks created by transit time uncertainties in less-than-truckload (LTL) line-haul operations. Transit time uncertainty may undermine the performance of the load plan, which specifies the route for each shipment and is synchronized to reduce line-haul costs. In our study, risk assessment of a load plan is performed via a dispatch simulation under randomly generated travel time scenarios. The risk consequence is measured by the average excess operational

cost, including transportation cost and handling cost. Compared to existing line-haul network models embedded within integer programming approaches for load plan optimization, the dispatch simulation can evaluate the performance of a load plan more realistically. In addition, a heuristic search algorithm based on multi-tree pivots is provided to obtain a cost-efficient load plan that is robust to transit time uncertainties.

Chapter IV presents methodology to assess the consequence of risks which arise from the intentional contamination of a food supply chain. Different from many risk management practices, the source of risk in this problem is an intelligent adversary, e.g., a terrorist group, who intends to deliver chemical or biological toxins to consumers using the supply chain. First, a general modeling scheme based on state-space models is provided to describe the dissemination of toxin across consumed products in a food supply chain. Then, a case study based on a representative liquid egg supply chain is presented. Based on the system model, a risk assessment for different supply chain designs is performed by simulation. Moreover, an in-depth analysis is conducted to determine the worst-case consequence given an intelligent attack considering the operational characteristics of the system. The worst-case consequence tool developed is designed to be embedded within any risk assessment approach.

CHAPTER I

INTRODUCTION

The past two decades have witnessed the increasingly important role that global supply chains play in the world economy. Due to the application of new technologies and better business practices, supply chains are becoming fast and cost-efficient, in order to better satisfy demanding customers and to withstand fierce competition from rivals. However, as supply chains become global and lean, with broader geographical coverage and lack of redundancy, they are exposed to many more risks than ever before. If a supply chain is disrupted by unexpected events, the stakeholders of the supply chain may suffer large losses.

For example, Hurricane Katrina devastated the transportation network around New Orleans in August 2005. The powerful storm and subsequent flood led to damage of roads and bridges, closure of airports and seaports, and blockage of the waterway of the Mississippi River. Apart from a death toll of 1836 people, the hurricane also caused a huge economic loss estimated to be over US$ 80 billion ([68]). More recently, in October 2012 Hurricane Sandy reminded us again of the risks posed by national disasters.

Similarly, supply chains may also be disrupted by accidents. A manufacturer relying on international suppliers may be impacted by an accident tens of thousands of miles away, if the risk is not properly handled. The North European cell phone producer Ericsson lost a staggering US$2.34 billion in 2000 because of its poor response to the shortage of a key chip, which resulted from a minor fire accident in a New Mexico plant of its supplier Philips ([56]). Meanwhile, Nokia, the main competitor of Ericsson, successfully contained the impact of the same disruption by actively

contracting with alternative suppliers.

Apart from accidents, supply chain risks may result from intentional adverse actions. During the lockout of west-coast ports from San Diego to Seattle by unionized port workers in the September of 2002, the economy suffered an estimated loss of one billion dollars per day, with factories closed, perishable cargo spoiling, and inventory stock-outs for retailers. ([27])

Besides natural disasters, accidents or intentional sabotage, some risks due to the inherent uncertainty in the supply chain operations, *e.g.*, uncertainty in shipping time, variability of customer demand, also need to be appropriately managed in order to improve customer service and business profitability. Instead of fire fighting after a supply chain is disrupted, which tends to always be costly, a more measured way is to handle risks using the framework of Supply Chain Risk Management (SCRM), which primarily includes identification, assessment, prioritization and mitigation of the risks in supply chains ([33]). As a Chinese saying goes, "Think of danger in times of peace; Preparedness averts peril."

1.1 Concepts of Supply Chain Risk Management

In this part, the basic concepts of supply chain risk management will be introduced. As mentioned earlier, supply chain risk management is a discipline of risk management which primarily includes the identification, assessment, prioritization and mitigation of risks in supply chains. In short, the process of supply chain risk management includes the following steps: (1) Potential supply chain risks are identified by a systematic investigation of all factors (sources) that might lead to risks; (2) The consequence of each identified risk and its corresponding likelihood are determined by analysis; (3) The risks are prioritized based on their consequence and likelihood; (4) For those risks that need to be addressed with action, mitigation strategies are developed in accordance to the objective of the supply chain. In particular, this thesis focuses on

providing problem-specific quantitative methods to measure risk consequences and to develop appropriate strategies for risk mitigation.

1.1.1 Risk Identification

In essence, *risk* is the potential for loss. A risk arises from some uncertainty, but not all uncertainties lead to losses. A source of risk refers to an uncertain variable (internal, external or environmental) that reduces outcome predictability ([33]). Although there are many ways of categorizing the sources of risk (e.g. [39], [26]), supply chain risks are generally grouped into three categories: internal risk, network risk and environmental risk, based on the sources of risk (see [33], [12] and [50] for details).

Environmental risk arises from uncertainty that affects all entities in the related market, *e.g.*, political, economic, technological uncertainties. The booming or shrinking of a particular industry may also be considered as an environmental risk source. A conflict in the Middle East may lift the oil price, which generally increases the operational cost of all market participants. During the financial crisis, the demand for new automobiles (and other consumer goods) dropped dramatically. Although the impact varied from company to company, almost all car manufacturers were affected by this environmental risk.

Network risk arises from the interaction between organizations within the supply chain. For example, a fire in a Philips plant disrupted component supply to Ericsson. The closure of a seaport may lead to rerouting and delay of goods and undermine customer satisfaction. The inaccurate forecast of customer demand may affect the inventory decisions of the supplier, which is known as the bullwhip effect. The impact of network risk spreads within the supply chain network.

Internal risk usually lies within the boundary of the organization under study. The risk may arise from the "hardware" or "software" of the supply chain. The former type of risk may be related to infrastructure (*e.g.*, plants, facilities) and assets (*e.g.*,

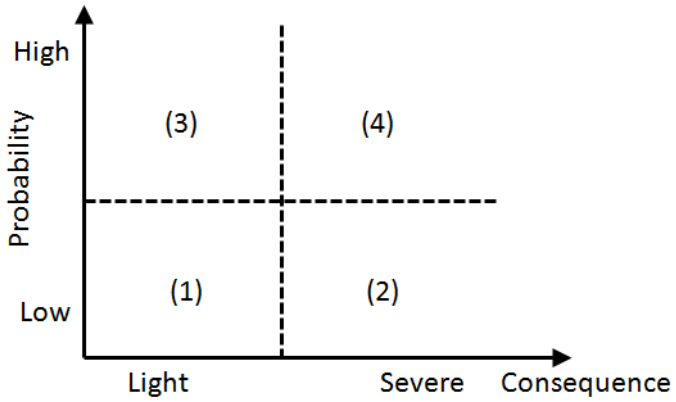

Figure 1: Dimensions of Risk Consequence

trucks, machines); the latter type of risk may be related to the controls that are used to guide supply chain decisions, *i.e.*, rules, procedures, assumptions, etc. Similarly, [12] calls the former type process risk and the latter type control risk.

1.1.2 Risk Measurement and Assessment

According to [34], risk can be represented by a triple: scenario (or outcome, or threat), consequence and likelihood. For each scenario, the consequence is the potential loss if the adverse event occurs; the likelihood is measured by the probability or frequency that the adverse event occurs. Assessment of consequences is usually performed by computational simulation or mathematical analysis. The likelihood is often computed based on probability theory or by using statistics methods. Since the assessment of risk is problem-specific, there is no general method for it. Thus, we illustrate the risk assessment by examples and analyze three typical problems in the following chapters.

After obtaining the profile of each risk, *i.e.*, consequence and likelihood, the prioritization of those risks can be visualized in a two-dimension chart, as illustrated in Figure 1 (adapted from [56]). The horizontal axis represents the magnitude of loss, while the vertical axis represents the probability that the adverse event occurs. A point in Figure 1 represents the profile of a particular risk.

The four quadrants in Figure 1 represent four type of risks: (1) low severity and low chance; (2) high severity and low chance; (3) low severity and high chance; (4) high severity and high chance. For example, the risk arising from breakout of Mad Cow Disease may be a first-type risk for a trucking company; the September 11 terrorist event belongs to the second type for an American airline; stock-out risk of an SKU for a Wal-mart store may belong to the third type; and the risk of power failure in a temperature-controlled warehouse may belong to the fourth type if the warehouse has no backup power, since the goods may perish without an air-conditioner.

1.1.3 Risk Mitigation Strategies

For each risk that needs to be handled, a proper strategy needs to be developed to mitigate the risk. In general, there are five types of risk mitigation strategies: (1) avoidance, (2) control, (3) cooperation, (4) flexibility, and (5) retention. See [33] and [39] for details.

Avoidance is a passive way to mitigate the risk by dropping or circumventing high-risk components of the supply chain, such as the market in a certain area, unreliable suppliers or carriers, or unprofitable products or customers. For example, piracy off the coast of Somalia is a threat to ocean vessels, and therefore simply avoiding shipping in this dangerous area, if possible, is an option to mitigate the risk. As another example, if a supplier has a bad record of on-time fulfillment, removing it from the list of suppliers may be an option to reduce risk in supply.

Control is a proactive way to mitigate risk by applying strategies to control contingencies, because simply avoiding the risk may not be an option in certain cases. For example, if there is variability in the delivery time of a supplier, then a proper inventory may be established to mitigate the risk of stock-out. For this problem, other control strategies may include vertical integration with the supplier or imposing contractual obligations on suppliers. Transferring the risk to an external entity,

e.g., an insurance company, is another way to control risk.

Cooperation is a proactive way to handle risk by joint efforts with other organizations on the supply chain. For example, the supplier may request that its customers to share more information about demand forecasts. By doing this, the bullwhip effect, *i.e.*, a specific risk of high inventory and shortage costs that results from excess demand variability, *i.e.*, may be reduced. In addition, establishing a way for more timely communication with customers may also mitigate this risk.

Flexibility increases responsiveness by allowing more degrees-of-freedom while leaving the predictability of uncertain factors unchanged. For example, postponement is a commonly used business strategy to mitigate the risk of demand uncertainty. Dell postpones the configuration of computers to the latest possible positions in the supply chain. By doing this, Dell takes advantage of a more accurate demand forecast for the aggregate demand of similar products. In general, flexibility may be considered as a special type of control strategy.

Retention is to recognize and accept the risk as a cost or constraint of operations. This strategy can be used for insignificant risk or a risk too catastrophic to be considered under usual conditions. That is why no insurance company provides an insurance product that hedges a risk related to wars, since the potential loss may be too huge to be covered by any premium. Those risks arising from catastrophic events are usually handled case by case.

To determine the appropriate risk mitigation strategy, a cost-benefit analysis is often performed. The first choice is to avoid the risk, ideally with a small cost. If it is impossible to avoid and we are unwilling to accept the loss, then developing a proper control strategy is a good option to mitigate the risk within the organization. In general, achieving operational flexibility is a good type of control strategy. If the risk needs to be handled with external organizations, a cooperation strategy may be deployed, if possible. But reducing the risk by cooperation may not work as well

as solving the problem within the organization. Finally, if no strategies work, then we have to accept the risk by considering it as a cost or constraint and need to be prepared to respond quickly when anything goes wrong.

1.2 Outline and Contributions

This dissertation develops contributions to the methodology for supply chain risk assessment and mitigation. In each of the three main chapters, a real problem is presented as an example of risk assessment and mitigation. The three prototype problems are chosen in such a way that: the first problem is about the infrastructure-related supply chain risk; the second problem is about control-related supply chain risk; the third problem is about human-related supply chain risk arising from intentional adverse actions.

Chapter II presents a methodology for assessing the risk arising from failure or malfunctioning of critical infrastructure and key resources (CI/KR) when a disruptive event occurs. The source of risk may be a natural disaster or intentional sabotage. Different from many risk assessment practices based on subjective evaluations, our methodology can systematically search for the most vulnerable supply chain components and measure their vulnerability (or risk consequence) by the supply chain cost impact created by the disruption. Modeling the flows on a supply chain network using a parametric minimum cost flow (PMCF) model, we analyze the impact of disruption by linear programming theory, and propose an efficient assessment algorithm based on the dual network simplex method. In general, the algorithm can assess the impact of any disruptive events having a single source of risk. Moreover, we extend the methodology and apply it to a time-space network so that vulnerability over time can be assessed. Finally, as a case study, the U.S. corn export supply chain network is assessed using a model constructed with public data sources. A risk assessment of the critical transportation network infrastructure (the dams and locks

on the Upper Mississippi River System) is performed, and a demonstration of how to identify the most vulnerable targets is performed. Different from other risk assessment methods based on qualitative ratings or subjective scores, our risk assessment methodology is based on the quantitative supply chain cost impact metrics; different from other risk assessment methods which analyze each target individually, our risk assessment methodology can systematically search for the most vulnerable targets in a large-scale supply chain network, and prioritize their risks. The primary contribution of this chapter is presenting a risk assessment and prioritization method, both in modeling and algorithm design, for infrastructure-related network risk.

Chapter III discusses the assessment and mitigation of risk arising from the uncertainty of transit time in less-than-truckload (LTL) line-haul operations. LTL carriers provide service to shippers (*i.e.*, customers) who tender shipments lower than the minimum truckload quantities, usually less than 10000 pounds. In order to increase the utilization of trailers and reduce the total operational cost, the LTL line-haul network is designed with a hub-and-spoke structure. Shipments collected by local operations are consolidated and sorted at terminals called end-of-lines, and sent to larger terminals called break-bulks, where the shipments are re-sorted. Usually, the routing of each shipment is controlled by a set of rules called load plan. Given the current terminal and destination of a shipment, the load plan specifies the next terminal the shipment is sent via a direct trailer move. However, uncertainty in transit times between terminals may undermine the performance of a load plan designed to work well when transit times are deterministic. In our research, the risk assessment of a load plan is performed via a dispatch simulation under randomly generated travel time scenarios. The risk consequence is measured by the average increase in total operational cost, including transportation cost and handling cost. Compared to standard integer programming models, the dispatch simulation can evaluate the performance of a load plan more realistically, by applying the control strategies used in practice.

In addition, a heuristic algorithm is developed to search for a cost-efficient load plan that is resistant to transit time uncertainties. In summary, the contribution of this chapter is threefold: (1) a method *i.e.*, dispatch simulation, is provided to realistically evaluate the performance of a load plan; (2) a risk assessment method is presented for the risk in the control strategy of the LTL line-haul operations; (3) and a robust load plan design algorithm is provided to obtain a cost-efficient load plan that is resistant to transit time uncertainties.

Chapter IV presents the assessment and mitigation of risk in a food production system under a potential intentional attack by inserting toxin agent into a system component. Different from many risk management practices, the source of risk in this problem is an intelligent adversary, *e.g.*, a terrorist group, who intends to attack the production system in order to cause morbidity and mortality. In order to describe the spread of agent in the system, a state-space model is built for each component in a production system. The assessment of risk given an attack is performed by running a simulation based on the system model. Then, an in-depth analysis is performed to determine the worst-case consequence given a smart attack considering the operational characteristics of the system. Finally, a model is proposed to determine the optimal design of the supply chain. The model optimizes system parameters in order to mitigate the risk while maintaining the system productivity. The chapter develops these approaches using a liquid egg supply chain as a test case study. Different from other methods which perform a high-level and qualitative risk assessment, our method provides a quantitative analysis on the operational level of the food production system. In addition, the risk is assessed on the assumption that the attacker is intelligent. The main contribution of this chapter is a risk assessment and mitigation methodology for risks arising from intelligent attacks in the liquid food production system.

CHAPTER II

VULNERABILITY ASSESSMENT ON SUPPLY CHAIN NETWORKS

2.1 Introduction

A supply chain network may encounter various events that disrupt its normal operations. The disruption may be a natural disaster, an accident or the result of intentional sabotage. For example, Hurricane Katrina devastated the logistics network around New Orleans in August 2005. Due to the powerful storm and subsequent flooding, many roads and bridges (*e.g.*, I-10 Twin Span Bridge) in that area were badly damaged. Airports and seaports were closed, and Mississippi River navigation was blocked by sunken barges. The logistics network in that area was severely disrupted, creating difficulties for the humanitarian relief and reconstruction work. Apart from the deaths of $1,836$ people, the hurricane also led to a huge economic loss that was estimated over US$ 80 billion ([68]). Another example of a major supply chain disruption occurred due to the longshoreman lockout at West coast ports from San Diego to Seattle in September 2002. The lockout led to an estimated loss of US$ 1 billion per day, with factories closed, perishable cargo spoiled, and inventory stock-outs for retailers. ([27]).

Disruptions in a logistics network are not necessarily limited to very large disasters. Any event that affects the normal or planned logistical operations can be considered a "disruption", *e.g.*, a stock-out of inventory, damage at a critical point on a transport route, the halt of a production system due to machine breakdowns, etc. If a logistics network is prone to disruptions that often lead to severe operational impacts, then the network is considered *vulnerable*. The vulnerability of a network is highly dependent

on the vulnerability of its most critical components; these components are often important flow *bottlenecks*.

To some extent, every logistics network is vulnerable. First, different parts of the network are interconnected and interdependent. Due to this inherent property of the network, a disruptive effect can be spread over the network and possibly be magnified during the spread. Second, logistics managers often design and control networks in the effort to make the system "lean" and cost-efficient by eliminating redundant resources and processes. However, this may increase the vulnerability of a system, since such redundancies can mitigate the impact of certain disruptive events. Third, most logistics networks rely on critical infrastructure and key resources. Damage to critical infrastructure and resources upon which the supply chain relies often leads to large system impacts.

In this chapter, we present a methodology useful for characterizing certain disruptive risks that potentially impact supply chain network operations. The methodology consists of a logistics network modeling approach, and algorithms useful for analyzing vulnerabilities. The method allows for a systematic search for the most vulnerable components in a network, and measures their vulnerability using a supply chain cost metric that varies given the magnitude of disruption. Given a particular magnitude of disruption, a ranking of network components by vulnerability is determined. The methodology is also developed for dynamic networks, and vulnerability of components over time can be assessed.

In our methodology, a logistics network is described using a Minimum Cost Flow (MCF) model ([1]), and the optimal flow in the network model is used to estimate the baseline supply chain cost for the subsequent vulnerability assessment. The optimal MCF flow is called the *nominal solution*. Then, a parametric model is built from the baseline model, where the magnitude of a potential disruption to a single supply chain network component is parameterized; the parametric model is used to estimate the

likely supply chain cost increase that results from a disruptive event. The vulnerability of a network component is described by an *impact curve* (IC), which represents the relationship between this cost impact and the magnitude of disruption. We prove that the impact curves that result from our models are piecewise linear and non-decreasing. By comparing their corresponding impact curves, the most vulnerable components of network can be determined.

To perform the vulnerability assessment, an algorithm based on network simplex is proposed. Given a disruption to a single network component, the algorithm determines each line segment that comprises the IC, by iteratively rerouting the flow from the disrupted component to alternate paths. The vulnerability assessment algorithm has two versions: static vulnerability assessment (SVA) and dynamic vulnerability assessment (DVA). The SVA is performed by directly applying the algorithm to the (static) network. In this case, the assessment reflects the instantaneous or average behavior of the network. The DVA is performed in a time-space network. In this case, vulnerability over time is assessed and the time periods at which the arcs are the most vulnerable is also determined.

The work described in this chapter makes several contributions. First, we propose an approach to model disruptions in logistics networks, and in particular develop a model for logistics networks facing disruption using a parametric MCF (PMCF) model. Second, we develop a framework for vulnerability assessment using both static and dynamic supply chain network models. Third, we use the models to study the relationship between the magnitude of disruption and the cost impact of disruption, and characterize its form. Fourth, we demonstrate the approach with a case study application to the U.S. corn export logistics network.

The remainder of this chapter is organized as follows. In section 2.2, the related literature is reviewed. In section 2.3, a methodology for vulnerability assessment is proposed using both static supply chain network models and dynamic supply chain

network models. For the static assessment approach, we model a logistics network under disruption as a Parametric Minimum Cost Flow (PMCF) model, where the magnitude of a disruption to a single network component is treated as a parameter. In this case, we show that the impact curve is a piecewise linear, non-decreasing function with respect to the magnitude of disruption. We also develop an algorithm for determining impact curves based on the dual network simplex method. For the dynamic assessment approach, a supply chain network model is built using a time-space network, and the key ideas of dynamic assessment are discussed. In section 2.4, the case study based on the U.S. corn export network is presented.

2.2 Related Literature

The earliest work focusing on vulnerability assessment for logistics systems dates back to the 1950s during the Cold War. The focus of this work is protecting against military attacks. A review in [29] summarizes the early work. The September 11 terrorist attack incident triggered renewed interest in quantitative vulnerability assessment, mainly focusing on intentional attacks by terrorists. Different industrial sectors in the U.S. have published their vulnerability assessment methodologies; see, for example, [4],[54],[38], and [64]. Hurricane Katrina in 2005 raised awareness of system vulnerability to natural hazards. Vulnerability assessment methodologies have been proposed for potential earthquake, hurricane or tsunami disasters, and are documented in [69].

Research has also focused on understanding supply chain vulnerability more generally, and has investigated the analysis of numerous risks facing supply chains. [43] presents a cross-sector empirical study of the sources and drivers of supply chain vulnerability. [55] presents a systematic discussion of the topic of supply chain vulnerability and develops the concept of resilient supply chains in order to address vulnerability problems. Neither of these references includes any quantitative models

to support decision-making for understanding or reducing supply chain vulnerability.

Vulnerability assessment is often embedded in the framework of risk assessment. [41] presents a good review of risk management in supply chains. The review surveys several papers in the literature that contribute to the four steps of risk assessment. The four steps are: (1) identification of key supply chain locations and threats, (2) estimation of probabilities and loss for each location, (3) evaluation of alternative countermeasures for each location, and (4) selection of countermeasures for each location. [70] proposes a disruption analysis network methodology based on a petri-net model, which helps identify how the effects propagate throughout a supply chain. [22], [40], [11] and [62] provide methods to identify potential threats. For estimating the probabilities of the disruptive events, [9], [51] and [36] focus on the application of expert systems and game theory for estimating risk of single catastrophes. They also suggest that probabilities of multiple catastrophes striking simultaneously may be estimated using simulation models. For analyzing the impact of disruptive events, [30] assesses the impact on the six types of key supply chain resources, including personnel, product/inventory, physical assets, public infrastructure, information, and financial.

A related area of study to the work developed in this chapter is the design of survivable networks. Survivable network design typically measures the vulnerability of a network using a metric of connectivity. The earliest work dates to the late 1960s (see, for example, [60]). Models developed with this goal seek to find a minimum-cost network topology that satisfies a connectivity requirement. Each time a direct link connection is established in the network, a nonnegative fixed cost is incurred. For the connectivity metric, an example is k-connectivity. For example, if a network is 2-connected, then, between any pair of nodes, there exist two different paths (more precisely, two edge-disjoint paths). If one path fails due to a disruption, the other path can be used to provide connectivity between the two nodes. [28] and

[49] provide comprehensive reviews on early papers on this topic. [10] focuses on 2-connectivity survivability and proposes heuristics for initial network construction and local improvement, while [25] proposes heuristics based on structural properties of the linear program relaxations. More recently, researchers have investigated polyhedral approaches and cutting plane algorithms for integer programming models of survivability. [5] presents a survey of the most recent work. Survivability models typically ignore arc capacity constraints or origin-destination flow requirements. Therefore, while they are useful for the analysis and design of communication networks, they may not be useful in supply chain applications. For example, if two nodes remains connected by a path that has little transportation capacity in a supply chain network, they are effectively cut off. Furthermore, it may not be possible to ensure that a pre-disruption feasible supply chain flow remains feasible after a disruption; it may be more important instead to ensure that important productivity metrics are not degraded substantially. To address these concerns, the method that we propose in this chapter models the supply chain as a capacitated network and performs analysis using capacitated network flow algorithms.

Researchers have also focused on determining the optimal allocation of defensive resources to a system in order to mitigate the risk of being disrupted. [7] gives a brief review for the literature in the area of defensive resource allocation. The resource allocation problems faced by the U.S. Department of Homeland Security (DHS) have often been the focus. Much of the work addresses a defender's optimization problem against exogenous attackers, usually intentional ones, *e.g.*, [8]. However, after Hurricane Katrina caused catastrophic devastation to a large region in the United States, researchers came to realize that protecting against natural (unintentional) disasters may require different resource allocations. Thus, [44] explores a way to strike a balance of allocating defensive investment between terrorism and natural disasters. [7] addresses a similar problem using a game theoretic approach.

The methodology we propose is based on parametric linear programming and the dual network simplex algorithm. The simplex method for un-capacitated networks is developed by Dantzig in [17], as a special case of the general simplex for linear programs. [18] provides an early discussion of a network simplex algorithm for capacitated networks. An excellent reference for theory and algorithms for the network flow problem is [1]. Closed related to our work, [58] and [59] develop a parametric analysis for the transportation problem, and use the framework to enable sensitivity analysis. The papers analyze the effect on an optimal solution if the parameters in the problem, *e.g.*, demand, arc cost, arc capacity, are slightly changed. Following this early work, [2] develops a re-optimization procedure for minimum cost flow problem based on network simplex algorithms and sensitivity analysis.

2.3 Methodology: Risk Consequence Assessment for Supply Chain Networks

Consider a logistics network, and suppose that the flow through the network can be modeled as a single product type, or commodity. Suppose that one can construct a model of the network that describes how and where the commodity is produced, stored, transported, and consumed over time. The methodology that we propose is to develop such a network model, where the commodity flows between production and consumption points can be predicted with sufficient accuracy by assuming that flows are allocated across available transportation and storage options to minimize total logistics cost. A simple network flow model of this type is the single-commodity minimum cost flow (MCF) model; see [1] for a detailed description. In an MCF model, arcs represent transportation and storage options, each with a capacity for maximum flow and an average cost per unit of flow.

Constructing an MCF model for a logistics network, and using an optimal flow to predict actual commodity transportation and storage choices and their concomitant costs may also allow for the analysis of disruptions to the network. In the event of a

disruption, *e.g.*, the temporary closure of critical infrastructure like a bridge or port facility, or the unavailability of a route like waterway or railroad, the arc flow costs may change and arcs may have capacity reduced or eliminated. In the event of a disruption, the logistics network will need to adjust (perhaps just temporarily) to the changed conditions. For example, flows may be delayed, cancelled, or may stored and transported using different options during a disruptive event. Such changes in flow will induce extra supply chain costs that than those of the optimal pre-disruption flow. The difference between post-disruption and pre-disruption cost is a measure of economic vulnerability.

If a disruption to some network component leads to a large cost impact, then we can categorize the component as vulnerable. The methodology we propose in this chapter can be used to identify and prioritize logistic network components using this measure of vulnerability. We also propose an approach where the logistics network is modeled explicitly over time, and this approach enables us to also identify the times of year when certain network components are most vulnerable.

We propose, then, two vulnerability assessment methods. The **Static Vulnerability Assessment (SVA)** uses a static network model of the logistics network under consideration, where flows represent time average flows (or total flows during an analysis horizon). The **Dynamic Vulnerability Assessment (DVA)** uses a time-expanded version of the static logistics network model to perform similar analysis. In this network, commodity flows on arcs represent total flows during short time periods, and individual network components are modeled with multiple arcs in the time-expanded network. The two versions are introduced in 2.3.1 and 2.3.2, respectively.

2.3.1 Static Vulnerability Assessment (SVA)

The proposed static vulnerability assessment (SVA) is based on a static logistics network model, in which flows represent average commodity flows (or total flows over a planning horizon). Since the network attributes are therefore time-invariant, the SVA considers the impact of a disruption in an aggregate way. For example, if the planning horizon considered is a year, the approach enables measurement of logistics cost loss incurred when the aggregate annual capacity of an arc is reduced or the average cost of an arc is increased.

In the remainder of this chapter, we will ignore disruptions that may impact the unit flow costs on arcs. It is possible to extend the analysis presented herein in a simple way to handle such disruptions, so we will focus instead on disruptions that may be modeled more appropriately as arc capacity reductions.

2.3.1.1 Nominal MCF Model

The nominal model characterizes the disruption-free behavior of the logistics network. Again, we focus on the case where we can develop a logistics network model where actual flows can be predicted accurately with optimal flows from a MCF model. The optimal flow obtained by MCF is called the nominal flow.

In a directed network $G = (V, E)$, V denotes the set of n nodes and E denotes the set of m arcs. Each node $i \in V$ may be associated with exogenous supply or demand of the commodity. Exogenous supply points are where flow enters the logistics network (production or procurement locations), and demand points are where flow exits the network (consumption locations). Parameter b_i denotes the commodity net supply of node i. If $b_i > 0$, node i is a supply node; if $b_i < 0$, node i is a demand node. There may also be logistics network nodes with balanced supply and demand, or nodes where no exogenous flow originates or is destined. For such nodes, $b_i = 0$ and node i is referred to as a transshipment node. For this work, we assume $\sum_{i \in N} b_i = 0$,

i.e., the supply and demand are balanced in the network.

Each arc $(i, j) \in E$ is associated with a coefficient c_{ij} that denotes the cost of unit flow sent via the arc, and a capacity u_{ij} that denotes the maximum allowable flow on the arc. The decision variables in the model are arc flows. Suppose that the flow on arc $(i, j) \in E$ is x_{ij}. Then, the nominal flow can be obtained by solving the MCF model below:

$$\text{Minimize} \quad \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i, \forall i \in N \tag{2}$$

$$0 \leq x_{ij} \leq u_{ij}, \forall (i, j) \in E \tag{3}$$

The objective function (1) is to minimize the total cost over all arcs; note that each arc cost is linear in flow, and that total cost is simply the sum of all individual arc costs. The constraints (2) are *balance constraints* that ensure that each node satisfies a flow conservation principle: the total outflow from the node minus the total inflow to the node is equal to its exogenous net supply. The constraints (3) define the capacity bound for each arc flow.

For convenience, we rewrite the above model in matrix form. Note that A is an $n \times m$ matrix, called the *node-arc incidence matrix*; [1] provides additional detail for standard MCF modeling constructs:

$$\begin{aligned} \text{Minimize} \quad & cx \\ \text{subject to} \quad & Ax = b \\ & 0 \leq x \leq u \end{aligned}$$

The MCF problem as written here is a linear programming problem. However, it is well known that for integer b, all of the extreme points of the constraint polytope are integer-valued. Thus, the problem of finding an integer flow x for an integer b is an example of an integer programming problem that is polynomially-solvable given

that linear programming is polynomial. Moreover, there are specialized polynomial algorithms with better worst-case complexity for solving MCF problems, such as the successive shortest path algorithm; see [1] for a complete review. In practice, these problems are typically solved with a simplex algorithm, such as the specialized network simplex algorithm for MCF or the more general dual simplex algorithm.

2.3.1.2 Parametric MCF model

Unexpected events, called disruptions in this research, may make the nominal flow impractical or even infeasible. Example logistics network disruptions may include the slowdown of production in a plant, the closure of an important bridge, etc. Although there are various kinds of disruptions, most of them lead to a similar logistics network impact: a reduction in the throughput or capacity of a logistics network component. Depending on how one models a logistics network, a disruption might cause a capacity reduction at a node or an arc. For example, a production slowdown at a plant might be thought of as a node capacity reduction, while a bridge closure would typically be modeled as an arc capacity reduction.

In general, we will model network components that have capacity as arcs, and therefore all disruptions will be arc capacity reductions. We summarize this idea with the following simple assumption:

Assumption 1 (Effect of a Logistics Network Disruption) *A disruption reduces the capacity of certain arcs in the network.*

For example, a plant can be modeled as two nodes connected by a single arc, where the arc represents the actual production capacity (and cost). Figure 2 (adapted from [1]) illustrates how this transformation is made in general. The transformation splits a capacitated node (left) into a pair of uncapacitated nodes (right), connected by a capacitated arc. Suppose the node capacity is u_n and its exogenous supply is b. Let $b^+ = \max\{b, 0\}$ and $b^- = \min\{0, b\}$. The input node, marked by a minus sign $-$, is

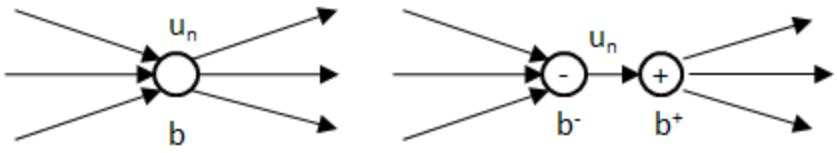

Figure 2: Network Transformation–Splitting of Node

given a supply of b^- and the output node, marked by a positive sign $+$, is given a supply of b^+.

It is rare that two or more independent large-scale disruptions happen simultaneously. Thus, in this research we assume that a logistics network faces only a single disruption at any given time. However, it is possible that a single disruptive event may impact multiple logistics network components simultaneously and synchronously. Thus, we allow a single disruption to impact the capacity of multiple network arcs in a synchronized way. We again summarize this idea with a simple assumption:

Assumption 2 (Single Disruption) *In the logistics network, there is at most one independent disruption impacting capacity at any given time.*

Consider now a disruptive event that may impact the capacity of one or more network components. In this research, we introduce a mechanism for modeling the capacity impact of such an event in a parametric way. Let θ be a scalar parameter measuring the impact of a disruptive event. If $\theta = 0$, the event does not impact the capacity of any network components. As θ grows, we assume that some component capacities will decrease.

For simplicity in this research, suppose that as the magnitude of a disruptive event increases, that some subset of the network arc capacities will decrease linearly in θ. Let $E' \subset E$ be the set of logistics network arcs affected by a specific disruptive event. For each arc $e \in E'$, let $d_e \in (0, 1]$ represent the impact of the disruption on arc e, such that an event with magnitude θ would reduce arc e capacity to $u_e - \theta d_e$. We

denote by d the vector $\{d_e\}$ for all arcs $e \in E$, where those that are not affected by the event have $d_e = 0$; we refer to d as the *disruption pattern* for an event. The most natural approach to build a disruption pattern for an event would be to first identify the arc (or arcs) for which the disruptive event would lead to the largest reduction of capacity, and for this arc (or arcs) set $d_e = 1$. Values for the remaining arcs impacted by the event would be scaled relative to this maximum.

Given this approach, we now propose a MCF model for predicting post-disruption logistics network flows. Given a disruptive event and a disruption pattern d for the event, we formulate the following parametric MCF model to predict post-disruption flows and costs where disruption magnitude θ is variable:

$$\begin{aligned} \text{Minimize} \quad & cx \\ \text{subject to} \quad & Ax = b && (4) \\ & x \leq u - \theta d && (5) \\ & x \geq 0 && (6) \end{aligned}$$

The only difference between the above model and the nominal MCF model lies in constraints (5). They reflect the predicted capacity of each arc given a disruption with pattern d and magnitude θ. In the remainder of this chapter, we refer to this model as the parametric MCF model (PMCF).

2.3.1.3 Impact Analysis

The supply chain cost impact of a disruptive event can be measured by the increase in total logistics costs, predicted by a logistics network model, given the disruption. First, let an optimal pre-disruption flow be denoted x^*, where this flow optimizes the nominal MCF problem. Furthermore, let z^* be the nominal MCF objective function (1) evaluated at x^*. Let $z(\theta)$ be the optimal objective function value of the PMCF

model for a given disruptive event with magnitude θ. Finally, then, let $z(\theta) - z^*$ measure the impact of the disruptive event. Note that for any $\theta \geq 0$ and $d \geq 0$, $z(\theta) \geq z^*$ since constraints (5) only contract the feasible space from that of MCF.

Since z^* is a constant, we can characterize the impact of disruptive events by characterizing the function $z(\theta)$. To do, suppose first that we restrict θ within a bound so that none of the arcs have capacity reduced to zero in the PMCF formulation. Specifically, the bound is $\theta \leq \min\{u_{ij}/d_{ij} \mid d_{ij} > 0, \ \ (i,j) \in E\}$.

We now prove three properties that characterize the function $z(\theta)$. For a fixed θ, the feasible region of the PMCF model is a polyhedron, *i.e.*, $P(\theta) = \{x \mid Ax = b, \ \ 0 \leq x \leq u - \theta d\}$. Since every feasible solution $x \in P(\theta)$, then, the optimal value is $z(\theta) = \min\{cx \mid x \in P(\theta)\}$.

The first property establishes convexity of the impact cost function in θ:

Lemma 2.3.1 *The optimal value $z(\theta)$ is a convex function of θ.*

Proof: Suppose x_1 and x_2 are optimal solutions of PMCF when $\theta = \theta_1$ and $\theta = \theta_2$, respectively. Also, suppose the corresponding optimal objective function values are $z(\theta_1) = cx_1$ and $z(\theta_2) = cx_2$.

Since x_1 and x_2 are feasible solutions, then $Ax_1 = b$, $Ax_2 = b$; and $0 \leq x_1 \leq u - \theta_1 d$, $0 \leq x_2 \leq u - \theta_2 d$.

For any $\lambda \in [0,1]$, we have $A(\lambda x_1 + (1-\lambda)x_2) = \lambda b + (1-\lambda)b = b$. Furthermore, $\lambda x_1 + (1-\lambda)x_2 \leq \lambda(u - \theta_1 d) + (1-\lambda)(u - \theta_2 d) = u - (\lambda\theta_1 + (1-\lambda)\theta_2)d$; and $\lambda x_1 + (1-\lambda)x_2 \geq 0$.

Suppose now that $x = \lambda x_1 + (1-\lambda)x_2$ and $\theta = \lambda\theta_1 + (1-\lambda)\theta_2$. Then, $x \in P(\theta)$, *i.e.*, x is feasible but not necessarily optimal solution to the PMCF minimization problem given θ.

Since the problem is a minimization problem, then the optimal solution objective function value $z(\theta) \leq cx$, thus, $z(\theta) \leq cx = \lambda cx_1 + (1-\lambda)cx_2 = \lambda z(\theta_1) + (1-\lambda)z(\theta_2)$.

Hence, the optimal cost $z(\theta)$ is a convex function of θ, by the properties of convex functions.

Q.E.D.

The second property shows that the convex function of θ is piecewise linear:

Lemma 2.3.2 *The optimal value $z(\theta)$ is a continuous, piecewise-linear function of θ.*

Proof:

Continuity of $z(\theta)$ follows directly from convexity. Now, consider the dual problem of the PMCF model:

$$\begin{aligned} \text{Maximize} \quad & b\pi - (u - \theta d)\alpha \\ \text{subject to} \quad & \pi^T A - \alpha \leq c \\ & \alpha \geq 0 \end{aligned}$$

where π is a vector of dual variables, corresponding to the constraints (4); and α is a vector of dual variables, corresponding to the constraints (5).

In practice, the nominal MCF is designed in such a way that a feasible solution must exist; otherwise, it would be the case that there is either no feasible predictive logistics flow or only an unbounded flow, which is certainly not desirable. Therefore, we can assume that the primal problem has a finite optimal value $z(\theta)$, and furthermore that the dual polyhedron is bounded by the theory of duality. A bounded polyhedron is also called a polytope. We note that θ does not appear in the definition of the dual polyhedron.

By the representation theorem, we know that the dual polytope can be represented by a convex combination of the extreme points. Suppose that the dual polytope has N extreme points: $(\pi_1, \alpha_1), (\pi_2, \alpha_2), \ldots, (\pi_N, \alpha_N)$, then, by strong duality,

$$z(\theta) = \max_{i=1,2,\ldots,N} \begin{bmatrix} \pi_i \\ \alpha_i \end{bmatrix} \begin{bmatrix} b & (\theta d - u) \end{bmatrix}$$

Given θ, the optimal solution of the dual problem (π_i, α_i) can be determined by simply selecting the appropriate extreme point. Furthermore, (π_i, α_i) determines the slope of a linear function of θ for each extreme point: for a fixed (π_i, α_i), $z(\theta)$ changes linearly with θ.

Since the number of extreme points is finite, as θ is varied the slope of $z(\theta)$ can change only a finite number of times, and thus the function contains a finite number of line segments. Thus, the optimal cost $z(\theta)$ is a piecewise linear function of θ.

Q.E.D.

The third property is that $z(\theta)$ is non-decreasing for θ small enough such that no arc capacities go to zero.

Lemma 2.3.3 *The optimal value $z(\theta)$ is a non-decreasing function of θ.*

Proof: Suppose $\theta_2 \geq \theta_1 \geq 0$ and x_1, x_2 are optimal solutions of PMCF when $\theta = \theta_1$, $\theta = \theta_2$, respectively.

Since $x_2 \in P(\theta_2)$, it follows that $x_2 \leq u - \theta_2 d \leq u - \theta_1 d$. Therefore, $x_2 \in P(\theta_1)$, with an objective function value of $z(\theta_2)$. Since x_1 is an optimal point in $P(\theta_1)$ with objective function value $z(\theta_1)$, it follows that $z(\theta_2) \geq z(\theta_1)$.

Hence, the optimal cost $z(\theta)$ is a non-decreasing function of θ.

Q.E.D.

We can now use the three lemmas to create a theorem characterizing the shape of the impact curve:

Theorem 2.3.1 *The function $z(\theta)$ is a convex, piecewise linear, non-decreasing function of θ, if $\theta \leq min\{u_{ij}/d_{ij} \mid d_{ij} > 0, (i,j) \in E\}$.*

Now we will consider how the function behaves each time some arc in the network has its capacity reduced to zero as θ increases. Define the set $L = \{u_{ij}/d_{ij} \ :| \ (i,j) \in E\}$. If $d_{ij} = 0$, then define $u_{ij}/d_{ij} = +\infty$. Let Θ_k be the k-th minimum value in the set L, then $\Theta_1 < \Theta_2 < \Theta_3 < \ldots$. Now, we will relax the assumption in Theorem 2.3.1 that $\theta \leq \Theta_1$. Suppose $\theta \in (\Theta_1, \Theta_2]$. Then, at least one arc will have zero capacity and thus zero flow. Using the network when $\theta = \Theta_1$ as the baseline and eliminating the corresponding variable x_{ij} by forcing its value to zero in PMCF, we obtain a new model. The impact curve of the new model still holds the properties in 2.3.1 with respect to $\theta' = \theta - \Theta_1$. Similarly, between any Θ_k and Θ_{k+1}, the impact curve is a convex, piecewise linear, non-decreasing function with respect to $(\theta - \Theta_k)$. Thus, the complete impact curve is characterized by the following theorem:

Theorem 2.3.2 *The optimal value $z(\theta)$ is a convex, piecewise linear, non-decreasing function between $[\Theta_k, \Theta_{k+1}]$, for all $k = 1, 2, \ldots$.*

The left part of Figure 3 provides an example of an impact curve as described by Theorem 2.3.1. The curve starts at coordinate $(0, z^*)$. As θ increases, the optimal value increases linearly at a higher slope when it passes breakpoints θ_1, θ_2, etc. The curve ends when $\theta = \Theta_1$; if this were the only breakpoint (for example, if $d_{ij} > 0$ for only a single arc (i,j)), the cost impact curve would be flat to the right, representing the intuitive notion that no additional damage can be inflicted on the network by increasing the disruption magnitude. Note that the first line segment may have a slope of zero due to slack capacity, indicating that this network component is immune to disruptions below a certain level of magnitude. The right part of Figure 3 provides an example of an entire impact curve. Between $[\Theta_k, \Theta_{k+1}]$, the part of curve is similar to the left curve. Again, to the right of the final breakpoint, the impact curve flattens to a horizontal line.

Given this characterization of the impact curve function, we now discuss computational issues for determining the curve. An optimal flow with respect to θ can

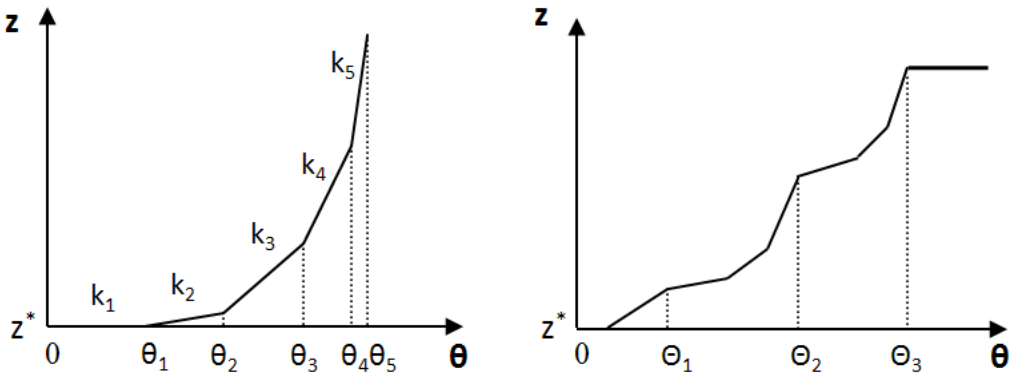

Figure 3: Example Impact Curves

be easily computed by relying only on solutions to the PMCF problem at potential breakpoint values Θ_k. Consider two adjacent breakpoints, θ_{i-1} and θ_i and a θ value between them: $\theta_{i-1} \leq \theta \leq \theta_i$. Suppose that optimal flows given θ_{i-1} and θ_i are x_{i-1} and x_i, respectively, with objective values $z(\theta_{i-1})$ and $z(\theta_i)$. Let $\lambda \in [0, 1]$ be such that $\theta = \lambda\theta_{i-1} + (1 - \lambda)\theta_i$. Since z is convex and piecewise linear with breakpoints only at values Θ_k, it is clear that $z(\theta) = \lambda z(\theta_{i-1}) + (1-\lambda)z(\theta_i) = c(\lambda x_{i-1} + (1-\lambda)x_i)$. If we define $x = \lambda x_{i-1} + (1 - \lambda)x_i$, we note that $z(\theta) = cx$. Furthermore, it is clear that $Ax = b$. To show that $x \leq u - \theta d$, we simply substitute and note that:

$$\lambda x_{i-1} + (1 - \lambda)x_i \leq \lambda(u - \theta_{i-1}d) + (1 - \lambda)(u - \theta_i d)$$

is satisfied due to the feasibility of x_{i-1} and x_i. Thus x is feasible and optimal for the PMCF problem given θ. Finally, we can also write the following expression for x and remove the intermediate computation of λ:

$$x = \frac{\theta_i - \theta}{\theta_i - \theta_{i-1}} x_{i-1} + \frac{\theta - \theta_{i-1}}{\theta_i - \theta_{i-1}} x_i$$

Given the impact curves for different network components, we can compare their cost vulnerability to disruption. To do so requires that we normalize the analysis, so that an identical value of θ yields a comparable magnitude of disruption or, perhaps

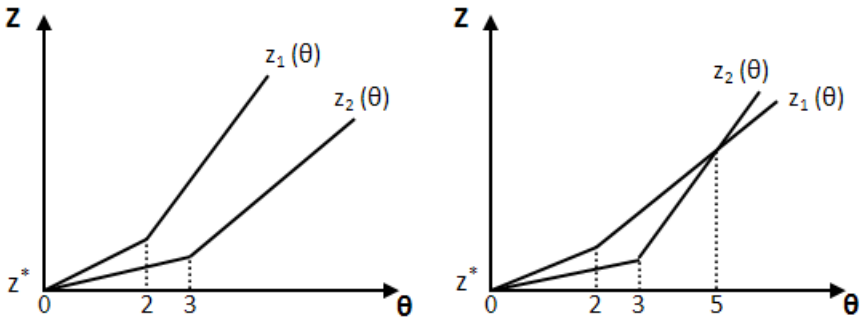

Figure 4: Two Sample Sets of Disruption Impact Curves

more appropriately, is representative of an equally-likely magnitude of disruption. Suppose we obtain such curves for two different network components. An example for what such curves might look like is given in Figure 4. In the left graph of the figure, the curve representing $z_1(\theta)$ is always higher than the one representing $z_2(\theta)$. Thus, for an equivalent disruption, the component corresponding to $z_1(\theta)$ suffers a larger cost impact than the one corresponding to $z_2(\theta)$; *i.e.*, component 1 is more vulnerable than component 2. In this example, the curve of $z_1(\theta)$ dominates the curve of $z_2(\theta)$. But in the right graph, no curve dominates the other. When the magnitude of disruption θ is small (less than 5), component 1 is more vulnerable; however, when the magnitude of disruption θ is large, component 2 is more vulnerable.

2.3.1.4 Static Assessment Algorithm

Now that we have characterized the disruption impact curve (IC) in 2.3.1.3, an algorithm is developed to determine an IC given a single disruption to a static network. In summary, the algorithm is an adaptation of the Dual Network Simplex method (see [1]). It builds the piecewise-linear IC function iteratively between successive breakpoints as θ increases, and is completed when θ is large enough such that each disrupted network arc (with $d_{ij} > 0$) has its capacity reduced to zero.

According to the characterization given by Theorem 2.3.2, an impact curve can

be represented by a list of breakpoint-slope pairs, denoted by

$$\{(\theta_1, k_1), (\theta_2, k_2), \ldots, \text{END/INF}\}$$

where θ_i is the i-th breakpoint and k_i is the slope of the line segment between θ_{i-1} and θ_i. Note that $\theta_0 = 0$, and that each impact curve begins at coordinate $(0, z^*)$ for a given logistics network model. Figure 3 provides an example to illustrate the representation. Every time a new breakpoint-slope pair is determined, it is pushed to the end of the list. When the algorithm terminates, if the network is still feasible, a special sign END is appended. If the network becomes infeasible, a special sign INF is appended.

To facilitate an understanding of the algorithm, we briefly introduce some background knowledge on the minimum cost network flow problem. Readers are referred to [1] for more details. The network simplex algorithm restricts its search for an MCF optimal solution to spanning tree solutions–the flow on any non-tree arc is fixed at its lower or upper bound. The flow variables associated with the tree arcs form a linear programming basis, thus we can refer to a spanning tree T as a basis. Given a flow x, the arcs in the network $G = (V, E)$ can be classified into three categories (L, T, U): $L = \{(i,j) \in E \mid x_{ij} = 0\}$; $T = \{(i,j) \in E \mid 0 \leq x_{ij} \leq u_{ij}\}$; $U = \{(i,j) \in E \mid x_{ij} = u_{ij}\}$. The dual variable associated with node $i \in V$ is also called a *node potential*, and is denoted by π_i. The reduced cost of arc $(i,j) \in E$ is defined by: $\bar{c}_{ij} = c_{ij} - (\pi_i - \pi_j)$. A solution, defined by the flow variables x and the dual variables π are known to be optimal if the flow is feasible (non-negative and satisfying the balance constraints) while: $\bar{c}_{ij} = 0, \forall (i,j) \in T$; $\bar{c}_{ij} \geq 0, \forall (i,j) \in L$; and $\bar{c}_{ij} \leq 0, \forall (i,j) \in U$.

Suppose that a disruption impacts multiple arcs simultaneously: let

$$S = \{e_1, e_2, \ldots, e_R\} \subseteq E$$

be the disrupted set. For each disrupted arc $e_r \in S$, let the disruption pattern coefficient be $d_r \in (0, 1]$. We now build up the algorithm recursively. Given a disruption of magnitude θ, let the current adjusted capacity of the arcs be $\hat{u} = u - \theta d$. Furthermore, let x be an optimal flow with respect to $\hat{u}$, T be an optimal basis, and π be optimal dual variables.

Suppose that θ increases and the capacity of disrupted arcs continues to reduce, then the flows on certain arcs may exceed capacity and need to be rerouted. For sufficiently small θ, it may be possible to reroute flow in excess of capacity on arc e_r in a way that does not require a change of basis T. Doing so is always the least-cost approach for flow re-routing given the capacity disruption, and the cost of the re-routing will be predicted directly by the current reduced costs.

Consider any arc $e_r \in S$. Let the tail of arc e_r be node i, and the head be node j. First, suppose that the basis T is not degenerate. If $x_r < u_r$, then $e_r \in T$ and an incremental decrease in the capacity u_r will not change the optimality of the flow x. Thus, no additional cost is incurred. Note, however, that since e_r is a tree arc, it may be used by other re-routing paths required by capacity reductions on other disrupted arcs. Suppose alternatively that $x_r = u_r$. An incremental decrease in capacity here now requires the excess flow to be re-routed. If the flow can be entirely re-routed feasibly using the unique directed augmenting path P_r from i to j using the tree arcs T, then the cost of the new solution will increase by $-cr$ multiplied by the capacity reduction. Finding the augmenting path in T from i to j is simple via breadth-first or depth-first search.

Now we consider the problem of jointly accommodating flow re-routing using tree arcs across the set of disrupted arcs S. Suppose that θ is increased by $\Delta\theta$. For each $r \in S$, the corresponding tree path is P_r; in case that $e_r \in T$, then $P_r = \{e_r\}$. Given the current capacity u_e of each arc, it is true that for each $r \in S$, the corresponding tree path P_r must accommodate the flow $\Delta\theta d_r$ (when $e_r \notin T$) or lose the same amount

of capacity (when $e_r \in T$). Therefore, to accommodate a disruption of magnitude $\Delta\theta$ without a basis change, the following feasibility condition must be met for each tree arc used by any re-routing paths:

$$0 \leq x_e + \sum_{p=1}^{R} \delta_{ep} d_p \Delta\theta \leq u_e, \forall e \in \cup_{r \in S} P_r$$

where δ_{ep} is the arc-path incidence indicator for arc $e \in E$ and path P_p for disrupted arc p:

$$\delta_{ep} = \begin{cases} 1 & e \text{ is a forward arc on } p \\ -1 & e \text{ is a backward arc on } p \\ 0 & e \text{ is not on } p \end{cases}$$

The summation $\sum_{p=1}^{R} \delta_{ep} d_p \Delta\theta$ accounts for the excess flows from all non-tree arcs p that are re-routed across $e \in T$; note that exactly $\Delta\theta d_p$ flow is re-routed, and if $\delta_{ep} = 1$ then this flow is added to the arc otherwise it is removed. If $p = e$, then $\delta_{ep} = 1$ by definition and $\Delta\theta d_e$ is exactly the capacity reduction due to the incremental disruption faced by this arc.

Thus, the current basis T can fully accommodate a disruption while this feasibility condition holds for all arcs e. We can use a ratio test to find the maximum value of $\Delta\theta$ such that these conditions hold. Denote $\gamma_e = \sum_{p=1}^{R} \delta_{ep} d_p$ and

$$\Delta\theta_e = \begin{cases} \frac{u_e - x_e}{\gamma_e} & \text{if } \gamma_e > 0 \\ \frac{x_e}{-\gamma_e} & \text{if } \gamma_e < 0 \end{cases}$$

Then, we can determine the maximum $\Delta\theta$ by the minimum of these ratios: $\Delta\theta = \min \{\theta_e \mid e \in \cup_{r \in S} P_r\}$.

Given θ_i, we can increase the magnitude of disruption to $\theta_{i+1} = \theta_i + \Delta\theta$ without changing the current network simplex basis. Within this range, the slope of the impact curve line segment is $k_i = \sum_{r=1}^{R} \mid \bar{c}_r \mid d_r$. For example, suppose that the magnitude of disruption increases by 1. For each disrupted arc $e_r \in S$, if $e_r \notin T$ its

flow is reduced by d_r due to the capacity reduction, and the total cost is increased by $d_r \mid \bar{c}_r \mid$; and if $e_r \in T$, then the capacity decrease does not affect the total cost, and $\bar{c}_r = 0$. In both cases, the calculation of the slope is valid. Then $(\theta_i + \Delta\theta, k_i)$ is added to the list, if $\Delta\theta > 0$. If $\theta = 0$, degeneracy occurs, which will be explained later.

At this point, we can update the network capacities and the optimal flow according to the re-routing plan. Increasing the magnitude of disruption beyond θ_{i+1} cannot be accommodated within the current basis, and thus the basis will need to be changed (along with the corresponding node potentials and reduced costs) to accommodate any additional disruption.

To perform a basis change, an arc to leave the basis needs to be selected first, according to a pivot rule from the bottleneck arcs which first reach their upper (u) or lower bound (0) after updating flow and capacity; note that continuing to augment flow would send such arcs into infeasibility. (Readers interested in pivot rules are referred to [6].) The removal of the leaving arc separates T into two sub-graphs T_s and T_t and forms a cut $(H, \bar{H})$. If the leaving arc (s, t) reached its lower bound (such that continuing flow augmentation would render its flow negative), then H is the set of nodes connected via T to t; in this case, we seek additional flow augmentation from t to s. Alternatively, if (s, t) reached its upper bound, we seek to augment additional flow from s to t and thus H is the tree of nodes connected via T to s.

To determine the arc (i, j) to enter the basis, we first note that it must belong to the cut $(H, \bar{H})$, since the addition of the arc must form a new tree. Second, (i, j) must have residual capacity for augmenting flow from H to $\bar{H}$. Residual capacity represents either flow that can be removed from $\bar{H}$ to H, or flow that can be added from H to $\bar{H}$. Thus, the arc (i, j) must be in the eligible set $Q = ((H, \bar{H}) \cap L) \cup ((\bar{H}, H) \cap U)$. Third, (i, j) must have the lowest absolute reduced cost value $\eta_{ij} = \mid \bar{c}_{ij} \mid$ among the arcs in Q.

To perform the pivot, we note that after (i, j) joins the basis, its reduced cost must

be zero: $\bar{c}_{ij} = 0$. By reducing the dual potentials of all nodes in $\bar{H}$ by the same amount η_{ij}, $\bar{c}_{ij}$ is driven to zero. Since $\pi_k \leftarrow \pi_k - \eta_{ij}$, it is clear that $\bar{c}_{ij} \leftarrow c_{ij} - \pi_i + \pi_j - \eta_{ij} = 0$ if $(i,j) \in L$ and that $\bar{c}_{ij} \leftarrow c_{ij} - \pi_i + \eta_{ij} + \pi_j = 0$ if $(i,j) \in U$. Now consider each arc $(k,\ell) \in (H,H) \cup (\bar{H},\bar{H})$; the reduced costs of these arcs do not change after the potential update, and thus basic arcs remain at reduced cost, and arcs at their lower and upper flow bounds respectively maintain non-negative and non-positive reduced costs respectively. Finally, consider the arcs in $(H,\bar{H})$. If the arc (s,t) that left the basis reached its flow upper bound, then $t \in \bar{H}$, and the new reduced cost $\bar{c}_{st} \leftarrow c_{st} - \pi_s + \pi_t - \eta_{ij} \leq 0$. Alternatively, if (s,t) left the basis to its lower bound, then $s \in \bar{H}$ and $\bar{c}_{st} \leftarrow c_{st} - \pi_s + \eta_{ij} + \pi_t \geq 0$. Each remaining arc in $(k,\ell) \in (H,\bar{H})$ was originally non-basic, but was not selected for the dual pivot. If $(k,\ell) \in L$, its original reduced cost was non-negative but not less than η_{ij}. The new reduced cost is either increased by η_{ij}, or reduced by η_{ij} leaving it to remain non-negative. If $(k,\ell) \in U$, its original reduced cost was non-positive but not more than $-\eta_{ij}$. Again, reducing this reduced cost by η_{ij} leaves it non-positive, as does increasing it by η_{ij}. Thus, the pivot preserves dual optimality. Given the new basis, additional flow re-routing is enabled for the disrupted arc set S.

Finally, there is a degenerate case to consider. It is possible to perform a pivot, only to find that further flow augmentation is limited to $\Delta\theta = 0$. In such cases, it is necessary to perform another pivot to change the tree structure. Degenerate cycling is thus certainly possible. However, such cycling can be prevented via selection of an appropriate pivot rule; this topic is beyond the scope of this document.

The above paragraphs describe an iteration of the algorithm. The algorithm starts with the nominal solution and repeats the iterations until one of two termination conditions is met. As the magnitude of disruption increases, the zero-capacity arcs are deleted from the set S. If S is empty, the algorithm terminates. In this case, END will be added to the description of the impact curve, indicating that there is no

Algorithm 1 Static Assessment Algorithm

Initialize by determining a nominal optimal network flow, basis and dual potentials;
while $S \neq \Phi$ **do**
  For each $e_r = (i, j) \in S$, determine tree path P_r from i to j;
  For each $e \in \cup_r P_r$, determine θ_e;
  $\theta = \min\theta_e$;
  if $\theta \neq 0$ **then**
    Decrease flows and capacity for all $e_r \in S$;
    Update flows for all $e \in \cup_r P_r$;
    Calculate the slope k and add (θ, k) to IC;
    Remove zero-capacity arcs from list S;
  end if
  Determine an arc to leave the basis and find cut $[S, \bar{S}]$;
  Determine the entering arc, if any;
  if (no entering arc is found) then add INF to IC and break;
  Update tree;
  Decrease the potential of nodes in $\bar{S}$;
end while
if $S = \Phi$, add END to IC;

additional marginal impact for larger values of θ. Another possibility is that during some iteration, a change of basis is required to augment additional flow, but $Q = \emptyset$. Since no improving basis exists, then no additional flow can be augmented and the problem becomes infeasible for larger values of θ. In such cases, we append INF to the description of the impact curve. Generally, it is better to develop a model that avoids such a possibility by penalizing flow that cannot move via the logistics network, since the notion of an infeasible flow is not clear in practice in this case.

The static assessment algorithm is summarized in Algorithm 1.

To illustrate the static assessment algorithm, Figure 5 gives a sample network with 8 nodes and 15 arcs. Suppose arc $(1, 2)$ and $(1, 3)$ are disrupted simultaneously, with $d_{12} = 1$ and $d_{13} = 0.5$. The upper left network gives the cost and capacity of each arc. The upper right network gives a nominal optimal flow. The dashed line highlights the tree. In the first iteration, the tree paths for arc $(1, 2)$ and arc $(1, 3)$ are $1 - 2$ and $1 - 4 - 3$, respectively. The first breakpoint $\theta_1 = \min\{\frac{20-15}{1}, \frac{80-25}{0.5}, \frac{40-25}{0.5}\} = 5$. The slope $k_1 = 1 * 0 + 7 * 0.5 = 3.5$. After updating the flow and capacity, the arc

$(1,2)$ leaves the basis, resulting in a cut with $S = \{1,3,4,5\}$. The set of eligible arcs $Q = \{(3,2),(5,7),(5,6)\}$. Since arc $(5,6)$ has the minimum $\eta_{56} = 1$, it is the arc entering the tree. After updating the basis and decreasing the potential of nodes in $\bar{S}$ by η_{56}, the lower left network results. In this iteration, the tree paths of for arc $(1,2)$ and arc $(1,3)$ are $1-4-3-5-6-8-7-2$ and $1-4-3$, respectively. The second breakpoint $\theta_2 = 0$, since arc $(3,5)$ is saturated and has no capacity for augmenting flow. This is a degenerate basis. After arc $(3,5)$ leaves the basis, arc $(3,2)$ enters the basis. The lower right network demonstrates the next iteration. The tree paths for arc $(1,2)$ and arc $(1,3)$ are $1-4-3-2$ and $1-4-3$, respectively. The third breakpoint $\theta_3 = 25/3 = 8.33$ and $k_3 = 8*1+7*0.5 = 11.5$. Arc $(4,3)$ leaves the basis. The corresponding cut is $S = \{1,2\}$. However, no eligible arc can be found; thus, the network becomes infeasible if θ continues to increase. In summary, the IC for this disruption pattern to arc $(1,2)$ and $(1,3)$ is $(5,3.5),(8.33,11.5),\text{INF}$. Using a similar method, we can also obtain the impact curves when the two arcs are disrupted individually. For arc $(1,2)$, the IC is $(5,0),(15,8),\text{END}$; for arc $(1,3)$, the IC is $(15,7),(5,8),\text{END}$.

2.3.2 Dynamic Vulnerability Assessment

2.3.2.1 Time-space network MCF Model

If we need to analyze the vulnerability of a logistics network over time, the assessment can be performed using a time-space network obtained by expanding a static network over time for a given planning horizon. Each time-space node (in the time-space network) represents a static node (in the static network) at a particular time. In this research, there are three types of possible time-space arcs: transportation arcs, inventory arcs and backorder arcs. Each *transportation arc* represents a static arc starting at a particular time. Inventory arcs only exist between time-space nodes corresponding to the same location that can hold inventory, connecting from one time period to the next one. If late delivery of the commodity is allowed, backorder

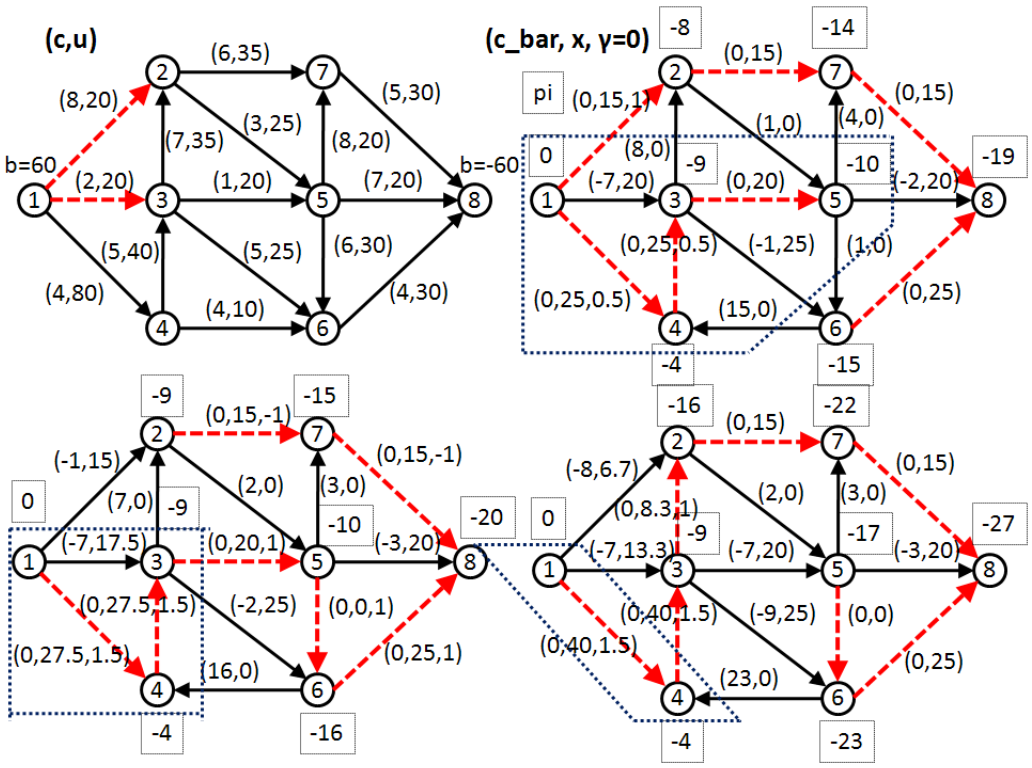

Figure 5: Static Assessment Algorithm Example Iterations

arcs may exist between time-space nodes corresponding demand points, connecting from one time period to the previous one. The flows on inventory arcs, backorder arcs and transportation arcs represent corresponding inventory levels, backorder levels and transported quantities, respectively. Since inventory and backorder costs are often naturally assumed to be linear in quantity-time, this modeling approach is useful for including such possible decisions and costs. An optimal flow in the time-space network based MCF problem (called TSMCF) minimizes total logistics cost, including transportation, inventory and backorder costs.

One approach for building a time-space network is to assume that the planning horizon consists of T decision points, equally spaced in time. For example, a decision point may exist each day of the year, or each week of the year. Suppose in a static logistics network $G = (V, E)$, the set of nodes that can hold inventory is denoted by $H \subseteq V$ and the set of demand nodes is denoted by $D \subseteq V$. Denote the time-space network with decision points from time 1 to T by $G_{1..T} = (V_{1..T}, E_{1..T})$. Then, $V_{1..T} = \{(i,t) \mid i \in V, t \in \{1,2,...,T\}\}$ and $E_{1..T} = I_a \cup T_a \cup B_a$, where I_a, B_a and T_a are the set of inventory arcs, backorder arcs and transportation arcs, respectively. Thus, $I_a = \{(i,t,i,t+1) \mid i \in H, t \in \{1,2,...,T-1\}\}$, $B_a = \{(i,t,i,t-1) \mid i \in D, t \in \{1,2,...,T\}\}$, $T_a = \{(i,t,j,t+t_{ij}) \mid (i,j) \in E, t, t+t_{ij} \in \{1,2,...,T\}\}$, where t_{ij} denotes the transit time from i to j. Figure 6 demonstrates a static network and a corresponding time-space network. All nodes in this example network can hold inventory. Backorder arcs do not appear in Figure 6. They would be represented by the reverse arcs of the corresponding inventory arcs at demand nodes.

Since a time-space network is also a general directed network, the TSMCF problem can also be solved efficiently using MCF algorithms. There are some important modeling details, however. First, time-space networks can suffer from so-called *end effects* where flows in the first set and last set of time periods may be distorted if initial conditions and end conditions are not well-modeled. Usually, one of two approaches

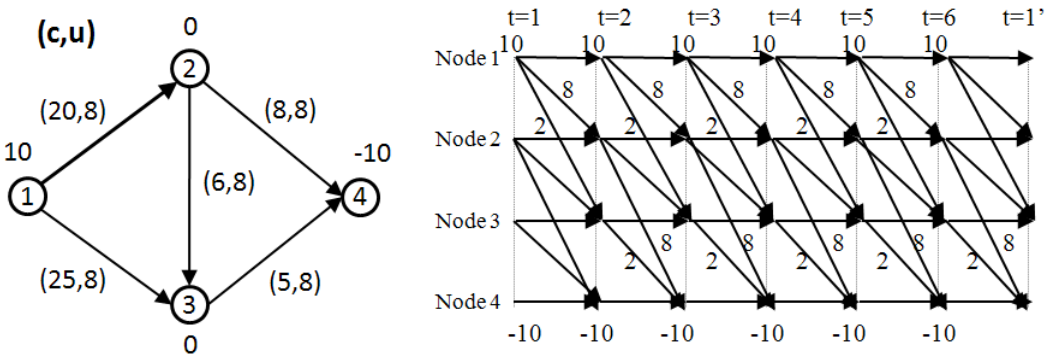

Figure 6: An Example of a Time-space Logistics Network

is used to address these end effects. If initial conditions can be determined with some accuracy, using a time-space network with a large enough value of T can mitigate the impact of end effects on flows early in the time horizon. Another option for periodic networks (with conditions that tend to repeat in cycles, like yearly flows) is to created a *wrapped* or *wrap-around* network where nodes near the end of the horizon are connected to nodes near the beginning of the horizon to model the periodicity; for example, inventory arcs connect nodes in period T to nodes in period 1. For example, consider the time-space network in Figure 6. A wrap-around version of this network is built by adding four arcs $(1, 6, 1, 0)$, $(2, 6, 2, 0)$, $(3, 6, 3, 0)$ and $(4, 6, 4, 0)$; wrap-around transportation arcs could also be added, but are omitted in the figure for clarity. For determining logistics network nominal flows, wrap-around networks are likely to be the most appropriate.

2.3.2.2 Modeling disruptions and recovery

We now consider how to model disruptions in a time-space network. First, time-space networks have a capability to model disruption to capacity and recovery of capacity over time. It should be clear that a disruptive event may initially have a strong impact on the capacity of a logistics network component, and that over time some or all of this capacity may be recovered. Figure 7 illustrates an example

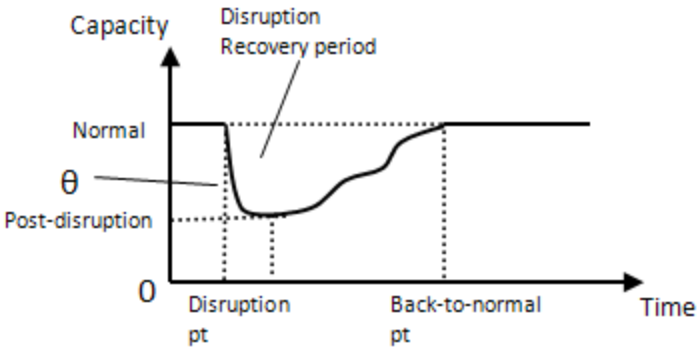

Figure 7: A Logistics Network Capacity Disruption and Recovery Over Time

disruption and capacity recovery process. The affected component is disrupted at the time labeled as the "disruption point". Initially, the impact of the disruption increases over time, and the capacity of the component drops to minimum (given a disruption magnitude). After that, the capacity reducing effect of the disruption diminishes and the component recovers slowly to the normal state at the time labeled "back-to-normal point".

Similar to the approach developed for the static network analysis, we can model the disruption in time-space network by appropriately setting the d vector for time-space arcs. For example, suppose the arc $(1, 2)$ and $(1, 3)$ are affected by the same disruption and the relative capacity reduction effect is twice as large for the former. Furthermore, suppose that the the effect of the disruption to $(1, 2)$ lasts from time 1 to 3 with relative magnitudes $0.9, 1, 0.5$ over the three time periods, until capacity is recovered in time 4; and the effect of disruption to $(1, 3)$ lasts from time 1 to 2 with relative magnitudes $0.8, 1$. Thus, the d vector components are then $\{0.9, 1, 0.5, 0, 0, \ldots\}$ for the corresponding time-space arcs of $(1, 2)$ and $\{0.4, 0.5, 0, 0, \ldots\}$ for the time-space arcs of $(1, 3)$.

2.3.2.3 Dynamic Vulnerability Assessment Framework

In general, we can assess the vulnerability of a time-space network by treating it as an ordinary network and applying the Static Assessment Algorithm (SAA) to it. However, there are two important modeling details that must be considered that both prevent the algorithm from "predicting" when a disruption will occur. First, it is critical that the first time period containing disrupted time-space arcs is $t = 1$. Second, the disrupted network must be modeled as a normal, non-wrap-around time space network. If these conditions are not satisfied, then the optimal flows on a disrupted network may seek to route flow around the disrupted network components in advance, and thus may predict the fact that a disruption is about to occur. Since we are primarily interested in unpredictable disruptions, this should instead be prevented. Note that some disruptions may be predictable; *e.g.*, forecasted natural disasters such as hurricanes.

In this research, we assume that we wish to understand vulnerabilities to unpredictable disruptions, where the time of a disruption occurring is not known in advance. Thus, we make the following assumption for DVA:

Assumption 3 (First Period Disruption Assumption) *The first disrupted logistics network arcs are modeled in the first period of a non-wrap-around time-space network for dynamic vulnerability assessment.*

To perform a dynamic vulnerability assessment under this assumption, it is necessary to extract a non-wrap-around sub-network from the entire wrapped logistics network model. The first time period for this sub-network is the first time period when the disruption affects the capacity of some network arc. Furthermore, suppose that the time window extracted for the sub-network contains r consecutive time periods after this initial time period.

The duration of the time window, r, is also called the *recovery period* in this

research. The duration should be chosen such that includes enough time for disrupted logistics network flows to recover to a pre-disruption normal state. Note that such a recovery will only occur if the logistics network completely recovers pre-disruption capacity at some point in time. We implicitly make this assumption in this research, but it is not necessary. Similar modeling techniques could be used if arc disruptions, represented by d, never return to normal (level 0); in this case, however, network flows would adjust to a new post-disruption steady state. Since this case is slightly more complicated to describe, we limit the discussion here to problems in which network components completely recover capacity.

Clearly, the time window duration r must be sufficiently long to ensure that each disrupted network component has completely recovered capacity. However, this duration may be too short to enable feasible flow recovery, since inventories and backlogs may have accumulated due to the disruption. It is also true that the shortest duration r that enables a feasible flow recovery plan may still not be the best choice, since a longer duration may enable recovery at a somewhat lower cost (and thus mitigate the impact of a disruption).

To extract a sub-network with duration r from a nominal wrap-around network, it is important to specify appropriate boundary conditions to account for flows that may be crossing the time boundaries of the sub-network. This is quite natural for the beginning of the time horizon; the reader should think about initial flows that may be in-transit at the time of the disruption, and are due to arrive at a somewhat later time to a specific node. But, it is also necessary for the end of the sub-network horizon. Boundary conditions can be specified using adjusted supply (or demand) for nodes that have flow from or to nodes that are outside the selected sub-network boundaries. These conditions will ensure that the flows outside the selected sub-network are the same as those in the original network after re-optimization given a disruption. The adjusted supply of each node in the sub-network is equal to the exogenous supply

in the nominal time-space network plus the optimal inflows from outside minus the optimal outflows to outside the sub-network.

Suppose the planning horizon is from time 1 to T and the sub-network contains r time periods. The original time-space network is denoted by $G_{1..T} = (V, E)$ and the sub-network from time t to $t + r$ is denoted by $G_{t..t+r} = (V_t, E_t)$. The subscript $t_1..t_2$ indicates that the time-space network includes all time-space nodes from time t_1 to t_2. Suppose the original supply of node $i \in V$ is b_i and its adjusted supply in $G_{t..t+r}$ is b_i^t. Then,

$$b_i^t = b_i + \sum_{(j,i)\in E\setminus E_t} x_{ji} - \sum_{(i,j)\in E\setminus E_t} x_{ij}$$

where $\sum_{(j,i)\in E\setminus E_t} x_{ji}$ is the optimal nominal inflow to node i from outside the sub-network and $\sum_{(i,j)\in E\setminus E_t} x_{ij}$ is the optimal nominal outflow from node i to outside of the sub-network. Using the adjusted supply b_i^t, the optimal flow on the complete nominal time-space network for the subset of arcs E_t remains optimal in sub-network $G_{t..t+r}$. To see why, it is first clear that the original optimal flow for the arc subset E_t is feasible for the sub-network by construction. Assume that an optimal flow on the sub-network is different from the original optimum, with a lower total cost. Then, replacing the flow on arcs E_t in the original network with the optimal sub-network flow will result in a strictly lower cost flow for the original network, which is a contradiction.

To apply the SAA on the sub-network, we need to reconstruct the basis T_s from the optimal flow in the sub-network, as well as determine the corresponding node potentials. According to the optimality conditions of MCF, all the arcs with $0 < x_{ij} < u_{ij}$ are in the basis. But this arc subset may not form a connected tree. Since more arcs are needed from $L \cup U$, we can arbitrarily add such arcs to T_s as long as no cycle is created, until there are $n-1$ arcs in T_s where n is the number of nodes in the sub-network. Given the basis T_s, node potentials π_i can be determined after setting

the potential of an arbitrary node to zero; since the reduced cost of basic tree arcs is zero, doing so simply scales the potentials around an arbitrary root node for the tree. Starting at this root, any tree search algorithm can be used to set the remaining potentials by recognizing that $c_{ij} = \pi_i - \pi j$ for all $(i, j) \in T_s$.

The Dynamic Assessment Algorithm (DAA) is now described, and is outlined in Algorithm 2. Note that if the complete time-space network contains wrap-around arcs, then the extracted sub-network may also wrap-around to the starting periods. For example, if $T = 5$ and $r = 3$, then time periods $5, 1, 2$ is a valid sub-network horizon, denoted by (5..8).

Algorithm 2 Dynamic Assessment Algorithm (DAA)

```
Obtain a nominal optimal flow in the original time-space network;
for t = 1 to T do
    Extract a sub-network G_{t..t+r} together with optimal flow x on its arcs;
    Assign the adjusted supply to each node in G_{t..t+r};
    Construct a basis tree T_s in G_{t..t+r};
    Determine potential π for each node i ∈ G_{t..t+r};
    Apply static assessment algorithm (SAA) on G_{t..t+r};
    Store the IC of the disruption in G_{t..t+r};
end for
```

To illustrate the vulnerability assessment in a time-space network, we will consider the network in Figure 6. The figure illustrates a nominal optimal solution for the complete time-space network. Suppose the inventory cost is 1 per unit of product per time period and the backorder cost is 10 per unit of product per time period. The disruption affects static arc $(1, 2)$, lasting from time period 1 to 3, during which the magnitude of disruption is constant. Thus, the components of d corresponding to the three time-space arcs are set to 1. Figure 8 demonstrates two key iterations in the assessment. In the first iteration, the disrupted flows are rerouted via the dashed tree paths, and $\theta_1 = 6$ and $k_1 = 6$. The reduced costs of three disrupted arcs are all -2. In the second iteration, the rerouting tree paths are represented by the dashed lines. Note that inventory arcs and backorder arcs are included in the tree paths,

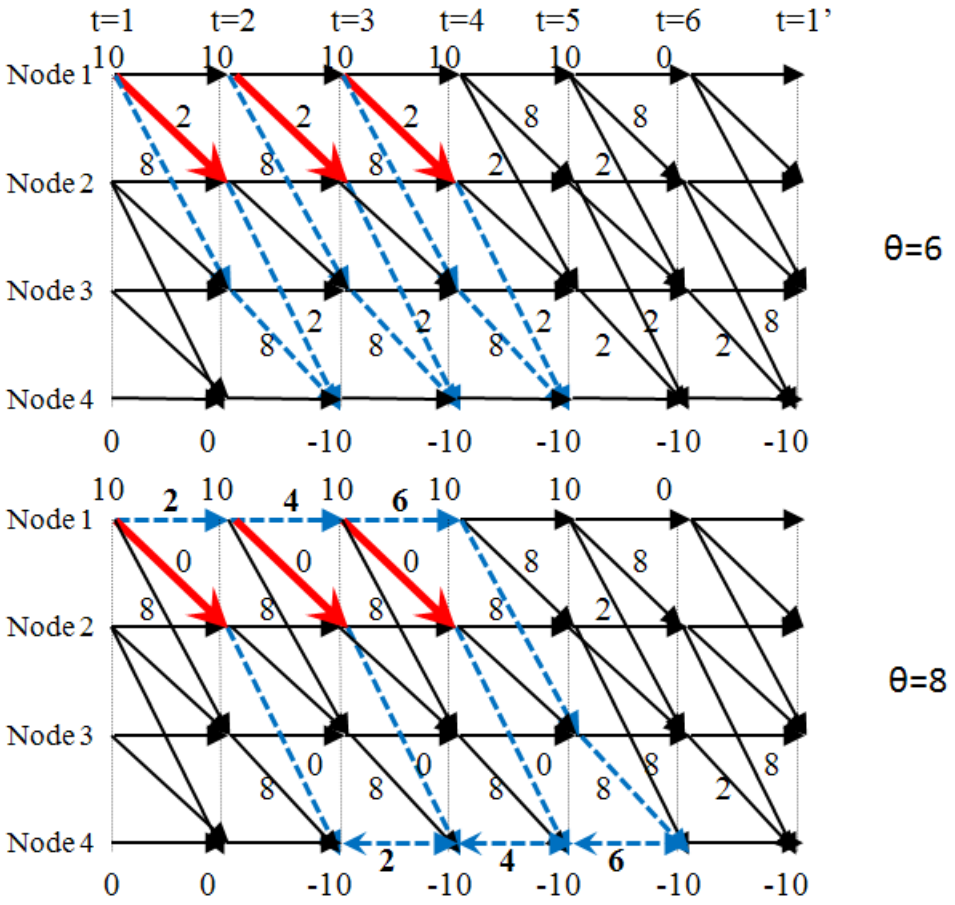

Figure 8: Dual Pivot Iterations on a Time-Space Network for the Dynamic Assessment Algorithm

which indicates that the disruption results in inventory holding as well as possible late deliveries. For this iteration, $\theta_2 = 2$ and $k_2 = 72$. Then the algorithm terminates. The final IC of this disruption is $\{(6, 6), (2, 72), \text{END}\}$.

Similarly, we can compute the IC if the same disruption occurs on the other three arcs. Note that since there is no optimal flow on arc $(2, 3)$, it cannot be affected by a disruption of any magnitude. The set of ICs is illustrated in Figure 9. By symmetry, the ICs of arc $(1, 2)$ and $(2, 4)$ are the same. So are the ICs of arc $(1, 3)$ and $(3, 4)$. Since the IC of the first two arcs dominates that of the latter two, arc $(1, 2)$ and $(2, 4)$ are more vulnerable to this type of three-period disruption.

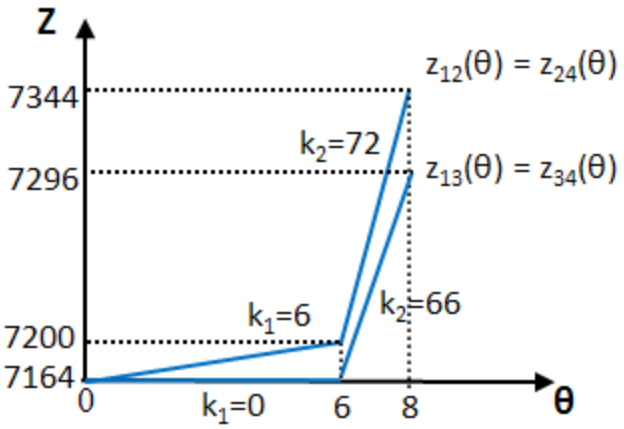

Figure 9: Result of the Dynamic Vulnerability Assessment for a Sample Network

2.4 Case Study: U.S. Corn Export Supply Chain

2.4.1 Introduction to Corn Export Supply Chain Network

The United States is the largest corn exporter in the world. According to the US Department of Agriculture (USDA), the total corn export volume was 56 million metric tons (mmt) in 2007, representing a 67% share of the world export of corn. In the United States, corn is mainly grown in the "corn belt", a region including Iowa, Indiana, Illinois, Ohio and parts of South Dakota, North Dakota, Nebraska, Kansas, Minnesota, Wisconsin, Michigan, Missouri, and Kentucky. The destinations of export flows inside the US are primarily Gulf of Mexico and Pacific-northwest (PNW) seaports. From these export ports, the grains are shipped to the destination country by bulk ocean vessel. Smaller export flows move to Canada and Mexico by railroad.

Within the US, export flows for grains rely on three freight transport modes– truck, railroad and inland water. Corn is shipped to New Orleans by barge using the Mississippi River system (Mississippi River, Illinois River, Ohio River, Missouri River, and Arkansas River) or by railroad; it is transported to PNW mainly by railroad. Usually, barge is preferred if available because of its low cost and large

capacity. From the perspective of capacity, a 100-car unit-train is equivalent to 384 trucks; a 15-barge flotilla is equivalent to 865 trucks. As a rule of thumb, truck transportation costs three times rail and rail costs three times barge ([66]). Corn held in inventory in this logistics network is stored in grain elevators of varying sizes, with capacities that range from 50 to 500 thousand tons.

In this case study, we focus on the long-haul transportation network that relies mainly on barges and trains to move grain from origin elevators in production regions to export seaports. Barge navigation of the Mississippi River system, and especially the upper Mississippi River, is enabled by a series of dams and locks. Our objective is to assess the vulnerability of the dams and locks if disrupted. The static and dynamic vulnerability assessment methodology will both be applied to the US corn export logistics network.

2.4.2 Logistics Network Model Development

The corn logistics network in the case study is built using geographic nodes that each represent a single Business Economic Area (BEA), a discretization of geography that is built around centers of economic activity. There are 179 current BEAs in the US, as defined by the US Bureau of Economic Analysis (see [32] for more information). The BEAs along the Mississippi River System and the railroads for corn export are represented by nodes in the network. The node supply (demand) is the total net amount of export corn produced (received) in that BEA. In cases where a single BEA is served both by the inland water system and the railroad system, two nodes are created: a *water* node and a *rail* node. Arcs between water nodes represent inland water transportation connections, and between rail nodes represent railroad transportation. Locks and dams on the inland water system are represented by arcs between water nodes, and additional water nodes are added as necessary to enable modeling these entities separately.

In our corn transportation network, most of the rail arcs are constructed according to the records of corn shipment in the Public Use Carload Waybill Sample collected by the US Surface Transportation Board (2007); see [61]. In addition, a number of important rail arcs, on which few corns are shipped under normal operating conditions, are added to provide some redundant capacity.

Now we consider the connection between the two transport modes: inland water and railroad. Since grains may be transported intermodally by railroad and inland water, we also model intermodal connections between water and rail nodes within the same BEA; such arcs represent the material transfer between the two modes. In this study, we assume that intermodal handling introduces no additional cost.

The parameters of the network are determined by data from different public sources and based on several assumptions. In the static network, the node demand and arc capacity are defined as total values for a single year. The annual supply (demand) of rail nodes and water nodes are determined from the railroad waybill mentioned above and the Grain Transportation Report compiled by the Agricultural Marketing Service of the US Department of Agriculture; see [63].

To determine the capacity of various arcs in the network, we follow an approach that attempts to provide an interesting case study example that is somewhat consistent with the real-world logistics system. First, using records from the Grain Transportation Report, we determine the actual annual corn flows that use each waterway arc. Since these are likely inconsistent with the supplies of nodes along the waterways, we adjust these supplies to be consistent by reducing the supplies on non-waterway nodes. At this point, we fix the capacity of the waterway arcs to be equal to these flows. Next, we use the waybill data to determine grain flows on the railroad network. Routes between origins and destinations are assumed to follow minimum cost railroad paths. Again, the arc capacities are set equal to these flows. Finally, certain railroad arcs are not used for any flows in this process. Such arcs are assigned capacity levels

that are similar to other nearby railroad arcs to provide some redundant capacity to the system. At the end of this process, the result is a waterway network with tight capacity and a railroad network with some capacity flexibility. Although the network may not be closely matched with the real capacities of the system, it should serve as a useful test bed.

For the transportation cost, we assume that water transport cost is 1.199 cents per ton-mile, and that rail transport cost is 3.556 centers per ton mile; these parameters are estimated by averaging the historical information provided in the Grain Transportation Report.

To create a logistics network for the dynamic vulnerability assessment, we expand the static network over 52 time periods, with each period representing a week. The transportation cost is directly obtained from the static network. Assume that every node can hold inventory, and that the annual inventory cost is estimated to be 20% of commodity value. Thus, the weekly holding cost is 50 cents per ton, given a price of corn at US$ 3.65 per bushel (based on 2007 figures in [63], noting that a bushel of corn is approximately 56 pounds). To avoid late shipments, we set the backorder cost to be US$ 2 per ton per week thus penalizing backorders at four times the rate of inventory holding. The weekly capacity of each arc is obtained by simply averaging the annual capacity in the static network. The supply (demand) of each node in the time-space network is estimated according to the weekly data and the seasonality of corn flow as reported in the Grain Transportation Report.

2.4.3 Computational Results

The algorithms proposed earlier are implemented in C++ with Microsoft Visual C++ 2008 Express Edition. All assessments reported in the following sections are performed on a DELL laptop with 1GB RAM and 2 Intel 1.6 GHz processors.

To perform the vulnerability assessment of dams and locks on Upper Mississippi

River System, we choose six representative dams and locks as potential disruption targets: Dam and Lock NO.15, NO.25, NO.27 on the Mississippi River, Dam and Lock NO.8 on the Illinois River, Dam and Lock NO.52 on the Ohio River, and Dam and Lock NO.1 on the Arkansas River. We will refer to these targets subsequently as MISS L15, MISS L25, MISS L27, ILL L8, OHIO L52 and ARK L1, respectively.

2.4.3.1 Static Vulnerability Assessment

In this example assessment, we disrupt the six targeted arcs individually and obtain the corresponding impact curves, illustrated in the left part of Figure 10. The right part of the figure displays the same curves, but only for low levels of capacity disruption. The impact curves of OHIO L52 and ILL L8 are the steepest. When their annual capacities are reduced by 2049 and 2261 thousand tons respectively, there is no longer a feasible logistics network flow, and the projected economic loss is approximately US $45 million. From an absolute perspective, OHIO L52 and ILL L8 are the most vulnerable targets in the static logistics network. However, neither dominates the other in vulnerability.

The impact curve MISS L25 is similar to that of OHIO L52 and ILL L8, which indicates that MISS L25 is also quite vulnerable to capacity reductions of similar magnitude. We now compare the vulnerability of MISS L25 and MISS L27. The two curves intersect when the magnitude of disruption is nearly 350 thousand tons. When the magnitude of disruption is below this value, the curve of MISS L27 is above MISS L25; when the magnitude exceeds this value, the curve of MISS L25 is above MISS L27. Thus, MISS L27 is more vulnerable for small disruptions while MISS L25 is more vulnerable for large disruptions.

MISS L15 is one of the least vulnerable targets. When the magnitude of disruption is below 2642 thousand tons, the flow on that arc is not affected. When the magnitude of disruption continues to increase, its impact curve has a small increase rate compared

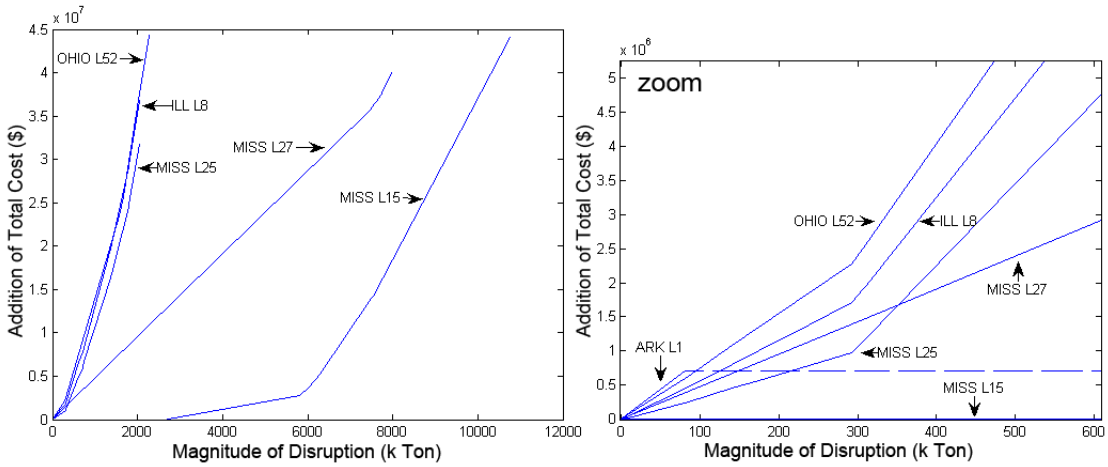

Figure 10: Impact Curves for Targets in Example Static Vulnerability Assessment

to the other targets. Another target with very low vulnerability is ARK L1. At first, its impact curve is steeper than all the other curves in the graph, but its disruption has a limited maximum economic impact. When its capacity is completely depleted, the total annual cost is US $0.7 million dollars and the logistics network flow remains feasible.

To interpret these results, we focus on two key ideas–redundant capacity and alternative routes. First, a target arc with redundant capacity can withstand small disruptions with no cost impact; the redundant capacity means that a capacity reduction does not change optimal network flows. For example, the optimal nominal flow through MISS L15 is below its capacity and thus this target is immune to disruptions with magnitude smaller than redundant capacity. All the other target arcs have tight capacity, so small disruptions will always lead to some cost impact. Second, the capacity and cost of alternative routes for a target arc also affect its vulnerability. If an arc has many low-cost alternative routes with available capacity, then even if it is severely disrupted, the flow can be rerouted without substantially increasing the total cost. In the corn network, MISS L25 and MISS L27 both have no redundant capacity. However, BEA 96 (St. Louis), on the upstream side of MISS L27, is a hub

for the railroad network. Rich railroad connections in BEA 96 provide good opportunities for rerouting the disrupted flow from MISS L27. That is why the impact curve of MISS L27 increases slowly. On the contrary, the alternative routes for MISS L25 do not have enough capacity for low-cost rerouting. Thus, the impact curve of MISS L25 becomes steep when the magnitude of disruption exceeds 300 thousand tons. Similarly, for OHIO L52 and ILL L8, a lack of good alternatives for rerouting leads to steep impact curves.

2.4.3.2 Dynamic Vulnerability Assessment

The transportation of corn is seasonal, *e.g.*, the fall and winter of the year are the peak seasons and the spring is a low season. In this part of the example vulnerability assessment, we perform a one-period disruption to each of the target dams and locks at different weeks during the year and obtain the corresponding impact curves. In Figure 11, we selected four weeks of the year for analysis, each during a different season: spring (week 14), summer (week 27), fall (week 40), and winter (week 52). Our intent is to compare the most vulnerable targets in four seasons of the year.

In all the four parts of Figure 11, the impact curves are all increasing with the magnitude of the disruption, until the point in time when a larger disruption cannot cause a larger impact. Since arcs are only disrupted for a single week in this example assessment, the logistics network flow never becomes infeasible (although in some cases, demand may be temporarily backlogged).

The results in the figure suggest that the vulnerability rank of the selected targets is generally consistent to the results of the static vulnerability assessment. For example, OHIO L52 and ILL L8 are the most vulnerable in most cases. But, the vulnerability ranks can be different from week to week.

Comparing the results for the four weeks, we observe that the supply chain in week 52 appears to be the most vulnerable, with the highest cost impacts. In this example,

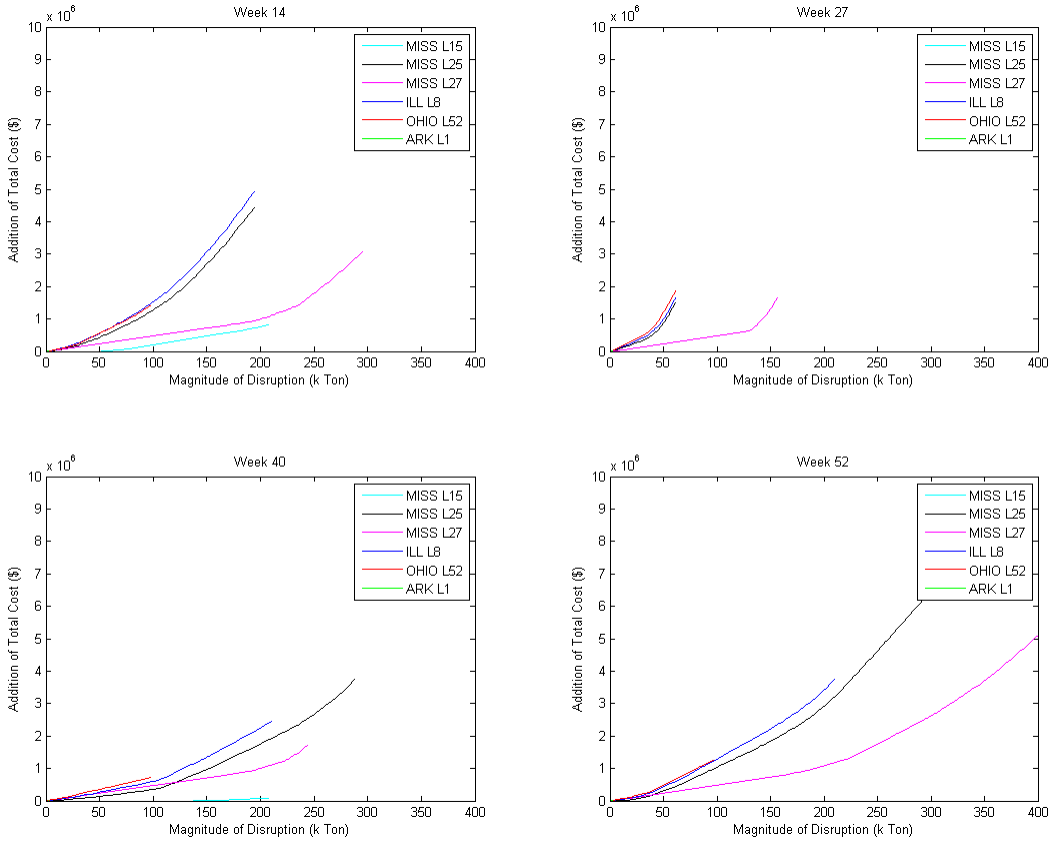

Figure 11: Impact Curves for Targets in Example Dynamic Vulnerability Assessment

Table 1: Most Vulnerable Locks and Time

Target	Disrupt Mag.(k Ton)	Target	Disrupt Mag.(k Ton)
ARK L1 (22)	$[1, 2]$	ILL L8 (17)	$[94, 150]$
OHIO L52 (20)	$[3, 38]$	ILL L8 (16)	$[151, 178]$
OHIO L52 (25)	$[39, 44]$	ILL L8 (14)	$[179, 193]$
ILL L8 (19)	$[45, 55][61, 63]$	ILL L8 (13)	$[194, 230]$
OHIO L52 (26)	$[56, 56]$	MISS L25 (44)	$[231, 244]$
OHIO L52 (27)	$[57, 60]$	MISS L25 (45)	$[245, 266]$
OHIO L52 (28)	$[64, 74]$	MISS L25 (46)	$[267, 321]$
OHIO L52 (29)	$[75, 78]$	MISS L25 (47)	≥ 322
ILL L8 (18)	$[79, 93]$		

we assume that the demand for corn transportation after the harvest season is high; alternatively, the summer is the least vulnerable season due to low transportation demand.

In addition, we also conduct an analysis to identify the most critical week for each target, *i.e.*, the target is most vulnerable to cost impacts if disrupted for that week. Table 1 lists the most vulnerable target-week combinations. The number in the parenthesis after the name of the dam and lock is the critical week. There are two peak seasons of demand, around weeks 30 and 45, which lead to busy agricultural transportation. The results in Table 1 indicate that the targets are vulnerable in peak seasons, which is quite intuitive. Moreover, the result of dynamic vulnerability assessment is consistent with that of static assessment. ARK L1 is vulnerable to small disruptions but has a limited maximum cost impact because of its small capacity. OHIO L52 and ILL L8 are again the most vulnerable, with the steepest impact curves. OHIO L52 is most vulnerable around week 27, since the harvest peak in its vicinity is around week 27. ILL L8 is most vulnerable around week 17 for similar reasons. Finally, MISS L25 is most vulnerable around week 45. MISS L15 does not appear in the list of most vulnerable arcs because its redundant capacity can absorb small disruptions. MISS L27 does not appear in the list since BEA 96 (St. Louis)

has many alternative railroad routes with available capacity to re-route flow diverted from MISS L27.

2.5 Conclusions

In this chapter, we present a methodology to perform a vulnerability assessment for system components in a supply chain network that face the possibility of severe capacity disruption. The challenge is modeling the disruption in a proper way and evaluating the impact of disruption effectively for a large-scale transportation network. By modeling the problem as a parametric minimum cost flow problem, we find useful properties of the impact curve that results from disruptions of increasing magnitudes, and develop an effective algorithm to perform an assessment based on those properties.

Although the methodology is designed to identify the most vulnerable system component in a supply chain network, the model can be generalized and used to identify the bottleneck of a general capacitated network useful for modeling in many application areas. For example, if the manufacturing processes of a plant can be modeled as a capacitated network, then the method can be applied to find the bottleneck process, the disruption of which will cause the most impact to the system performance.

CHAPTER III

ROBUST LTL LOAD PLAN DESIGN

3.1 Introduction

A load plan specifies how shipments are routed through a line-haul terminal network operated by a less-than-truckload (LTL) carrier; *i.e.*, for each shipment, given its origin, destination and customer service standard, a load plan determines the terminals at which the shipment is transshipped and thus the route of the shipment. In practice, a load plan usually conforms to a so-called *in-tree* structure; *i.e.*, the shipments destined to the same terminal flow into their common destination along terminal paths that form a directed tree rooted at the destination. At each terminal, the subsequent route of a shipment is determined by its destination, regardless of its origin. In this chapter, we assume that all load plans are comprised of directed in-trees, one for each destination.

Usually, an LTL shipment weighs between 150-10000 lbs, and uses approximately 5-10% of the trailer capacity. Thus, it may not be economical to send each LTL shipment directly from its origin to destination. Instead, at each terminal of the LTL network, the carrier consolidates shipments from different customers, loads them into an outbound trailer, and sends the trailer to their common next stop. Then, the shipments are unloaded from the inbound trailer, sorted, loaded into the outbound trailer, and shipped. Each shipment may be transferred several times before arriving at its destination.

In a line-haul network, each terminal serves as a hub of the city operations in the nearby areas. In the evening of each day (usually by 7 pm), the shipments collected by the pick-up vehicles of the city operations are consolidated at each terminal and

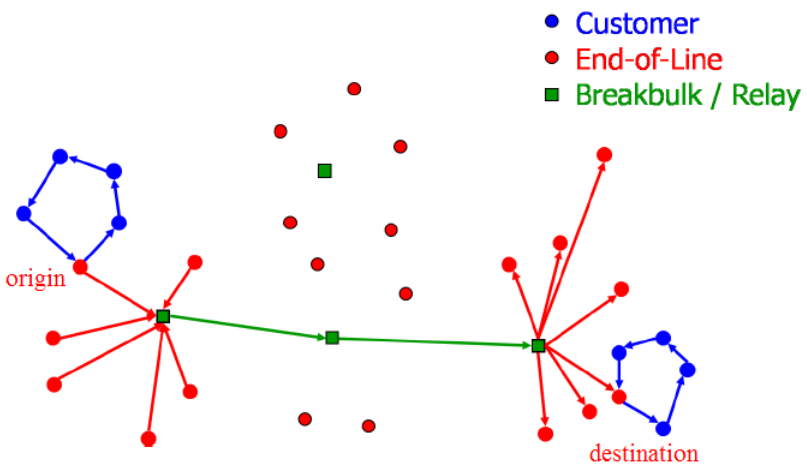

Figure 12: Structure of an LTL Network

loaded into outbound trailers for inter-city transportation. In the early morning of each day (usually by 8 am), the shipments arriving at the destination terminal are assigned to the vehicles of the city operations for local delivery. The due day on which the shipment is delivered is calculated based on the service level of the shipment. The terminals can be classified into two categories: end-of-lines (EOL) and break-bulks (BB). EOLs can only be the origin or destination of a shipment. On the other hand, BB terminals can additionally serve as transshipment points, where the shipments from the inbound trailers are unloaded, sorted and loaded into outbound trailers.

A *direct* is an important concept in load plan design. When a trailer travels on a direct, then the trailer will not be opened until arriving at the destination of the direct. During this transit time, however, the trailer may stop at some intermediate terminals for changing the driver, due to restrictions on the maximum working hours of a driver. Each path that comprises the load plan can be thought of as a sequence of directs connecting the terminals at which shipments are transferred.

Shipment consolidation increases the utilization of trailers and thus reduces total transportation cost. At the same time, the cross-docking operations requires additional material handling cost, and increases the transit time of the shipments because

of additional waiting and handling time at intermediate break-bulks. Moreover, consolidation increases the complexity of the transportation operations, and may introduce additional operational risks, which will be discussed later. Hence, the objective of load plan design is to minimize the total cost of line-haul operations, including the expected transportation cost and handling cost, while ensuring that the service requirement of each shipment is satisfied.

Line-haul operations involve a myriad of risks, which may undermine the performance of a load plan that works well under a disruption-free condition. For example, uncertainty in customer demand may result in the sub-optimality of a load plan obtained using a model that treats demand as a deterministic vector. Uncertainty in transit time may lead to delays of shipments, and the effect may spread to the downstream through the network. Hence, it is necessary to determine a robust load plan that performs well under both normal and disrupted conditions. In this chapter, we mainly consider the risk arising from the uncertainty in the transit time of a direct.

Trucks may be delayed for different reasons, *e.g.*, inclement weather, road congestion, traffic accidents, or an equipment breakdown. In a highly optimized LTL network that has the least redundant resources, the impact of even a single delay may be significant. Total operating cost may increase substantially, and late shipments may violate customer service standards. Thus, the risk in a load plan arising from transit time uncertainty needs to be assessed, and a method to obtain a robust load plan resistant to the risk needs to be developed.

3.2 Related Literature

The load planning problem is a type of network design problem, which has been extensively studied in the literature. For realistically-sized instances of the problem, it is often impractical to determine an optimal solution to the general network design problem by exact optimization. Heuristic algorithms are thus developed to find good

solutions. [15] proposes a tabu search algorithm using pivot-like moves in the space of path-flow variables, while [14] presents a similar scheme. [23] develops a tabu search algorithm using cycles that allow re-routing of multiple commodities. [24] integrates the above cycle-based neighborhood with a path-relinking algorithm. The general network design model assumes that shipments can be split among multiple paths, and load plan design typically prevents this possibility by enforcing the in-tree structure. Furthermore, the general network design model does not explicitly account for the resources used to provide transportation, and thus ignores issues like empty resource balancing.

The load plan design problem is more closely related to the service network design problem, but is still more specific. Early research on in-tree load plan design focuses on models that use static networks which do not explicitly capture service standard constraints or the timing of trailer dispatches. [45] provides a local improvement heuristic for solving a load plan design problem; in this work, fixed costs are introduced to the model by enforcing a minimum number of weekly trailer dispatches on any direct that is used to route any shipment flow, however fractional trailer flows above the minimum are treated as continuous random variables. Works related to this seminal study include [47], [48] and [46].

Unfortunately, models for load planning based on static networks tend to overestimate consolidation opportunities and thus may not provide a good estimate of the operational cost. [20] presents a model using a time-space network which is able to model the timing of consolidation. A heuristic algorithm is presented that determines arc sub-gradients in the service network by solving large-scale multi-commodity network flow problems. Again, shipments traveling from the same origin to the same destination can be split over multiple paths and the model does not consider empty equipment repositioning. [31] proposes a similar model but uses a simplified time-space network, *i.e.*, for each terminal there is only one time epoch each day. A column

generation approach is developed to create load plans, where a column represents a tree of shipment paths into a destination. In addition, both loaded and empty routing decisions are considered separately and sequentially. For a more comprehensive review of literature in this area, two excellent reviews are recommended: [13] and [67].

Most recently, [19] developed a detailed time-space network model for load planning, where transportation costs are linear in the integer number of trailers dispatched on every time-space direct. The proposed model uses a fine discretization of the time-space network to describe the consolidation opportunities more accurately. Empty trailers are also considered when decisions are made. The paper presents an integer programming-based local search method using both exact and heuristic search techniques to find a good load plan. The integer programming formulation used is a restricted version of the complete path-based load plan design formulation. Similar algorithmic approaches for different hard logistics optimization problems are proposed in [21], [52], [53] and [3].

3.3 Formulations of Nominal Load Plan Design

In this section, integer programming formulations of the nominal load plan design problem are provided, as a basis of the later discussions. The models proposed here are similar to those proposed in [19].

Suppose that a line-haul transportation network $G = (V, E)$ is given, where V is the set of terminals, including break-bulks and end-of-lines, and E is the set of potential directs connecting the terminals. Note that $\mid V \mid = n$ and $\mid E \mid = m$. For each terminal $i \in V$, the outbound directs are represented by $\delta^+(i)$ and the inbound directs are represented by $\delta^-(i)$. The handling cost per (fractional) trailer at terminal i is denoted by h_i. Assume the handling cost is proportional to the weight of the shipments transferred at the terminal. The transportation cost per (integer) trailer

on direct (i, j) is denoted by $c_{i,j}$. In our research, we assume $c_{i,j}$ is proportional to the mileage of the direct of (i, j).

In an LTL transportation system, a shipment can be represented by its origin o, destination d and weight measured by the fraction q_{od} of a trailer's capacity that is required to move the shipment. Note that $q_{od} = w_{od}/W$, where w_{od} is the weight of the shipment measured by pounds and W is the weight capacity of a truckload in pounds. q_{od} is not necessarily less than 1. According to the in-tree assumption of a load plan, at a given terminal the shipments are routed based on their final destinations, regardless of their origins. Thus, the shipments with the same destination will never be split once consolidated.

In a multi-commodity network flow setting, each commodity in the LTL network is defined by all of the shipments with the same final destination. A commodity can be denoted by the common destination of the shipments. Then, the set K of all commodities is a subset of the terminal set V. For simplicity, let $K = V$. The net supply of commodity k at terminal i is defined by b_i^k below. At origin i for commodity k, the net supply is equal to the fractional trailer quantity q_{ik}. At the common destination of commodity k, the net supply is the negative of the total net supply of all the other terminals. Note that for all commodity $k \in V$, $\sum_{i \in V} b_i^k = 0$.

$$b_i^k = \begin{cases} q_{ik} & \text{if } i \neq k \\ -\sum_{j \in V \setminus \{k\}} q_{jk} & \text{if } i = k \end{cases}$$

The load plan is represented by a set of binary decision variables $\{y_{ij}^k\}$. If $y_{ij}^k = 1$, then commodity k (destined for terminal $k \in V$) currently at terminal $i \in V$ is sent to terminal j as the next stop along direct (i, j). Otherwise, $y_{ij}^k = 0$. Note that $y_{ii}^k = 0$ and $y_{kj}^k = 0$, for all $i, j, k \in V$.

The multi-commodity trailer flow in the network is represented by the set of non-negative continuous decision variables $\{x_{ij}^k\}$. x_{ij}^k is the flow of commodity k on direct

$(i,j) \in E$, in units of fractional trailers. Note that no shipment will be split over different routes if flow x_{ij}^k is only allowed on directs (i,j) where $y_{ij}^k = 1$. Finally, τ_{ij} represents the integer number of trailers dispatched on direct (i,j), including both loaded and empty ones. The volume of shipments handled at terminal i, in terms of fractional trailers, is denoted by $z_i = \sum_{j \in \delta^+(i)} \sum_{k \in V} x_{ij}^k$.

An arc-based formulation of the load plan design problem in a static network is given as below:

$$\text{Minimize} \quad \sum_{(i,j) \in E} c_{ij} \tau_{ij} + \sum_{i \in V} h_i z_i \tag{7}$$

$$\text{subject to} \quad \sum_{j \in \delta^+(i)} x_{ij}^k - \sum_{j \in \delta^-(i)} x_{ji}^k = b_i^k, \forall i \in V, \forall k \in V \tag{8}$$

$$x_{ij}^k \leq M y_{ij}^k, \forall (i,j) \in E, \forall k \in V \tag{9}$$

$$\sum_{j \in \delta^+(i)} y_{ij}^k \leq 1, \forall i \in V, \forall k \in V \tag{10}$$

$$\sum_{k \in V} x_{ij}^k \leq \tau_{ij}, \forall (i,j) \in E \tag{11}$$

$$\sum_{j \in \delta^+(i)} \tau_{ij} - \sum_{j \in \delta^-(i)} \tau_{ji} = 0, \forall i \in V \tag{12}$$

$$z_i = \sum_{j \in \delta^+(i)} \sum_{k \in V} x_{ij}^k, \forall i \in V \tag{13}$$

$$x_{ij}^k \geq 0, \forall (i,j) \in E, \forall k \in V$$

$$y_{ij}^k \in \{0,1\}, \forall (i,j) \in E, \forall k \in V$$

$$\tau_{ij} \in Z^+, \forall (i,j) \in E$$

The objective function (7) minimizes the total operational cost, including the transportation cost linear in the trailer count variables and the handling cost linear in the flow variables. Constraints (8) ensure mass conservation of each commodity at each terminal. Constraints (9) ensure that the routes of each commodity are consistent with the load plan, where M is a sufficiently large integer. Constraints (10)

enforce the in-tree structure of the load plan. Constraints (11) are round-up constraints that ensure that there are enough trailers on each direct. Finally, constraints (12) ensure that trailers are balanced through empty trailer repositioning.

We now extend the formulation to a time-space network, which can be used more naturally to capture important flow timing costs and constraints. Suppose that the static network is $G = (V, E)$ and the time horizon is represented by a set of discrete time epochs $T = \{t_p\}$. Note that the time is ordered, *i.e.*, for all p, we have $t_p < t_{p+1}$. Let the corresponding time-space network be $G_T = (V_T, E_T)$. Then, $V_T = \{(i, t) \mid i \in V, t \in T\}$ and $E_T = I_a \cup T_a$, where I_a and T_a represent the set of inventory arcs and transportation arcs, respectively. Thus, $I_a = \{((i, t_p), (i, t_{p+1})) \mid i \in V \text{ and } t_p, t_{p+1} \in T\}$, and $T_a = \{((i, t_i), (j, t_i + t_{ij})) \mid i, j \in V \text{ and } i \neq j \text{ and } t_i, t_i + t_{ij} \in T\}$ where t_{ij} is the transit time from terminal i to j.

If we assume splittable commodity flows such that freight traveling from location i at time t due to arrive at location d at time t' can be dispatched at various time epochs en route, then we can extend the arc-based formulation of the load plan design problem directly. Suppose $(u, v) \in E_T$ is an arc in the time-space network, where u and v denote the time-space nodes serving as the tail and head of the arc. For (u, v), let the corresponding direct in the static network be denoted by $(i, j) = s(u, v)$. And $i = s(u)$, $j = s(v)$. Note that the function s is a mapping from time-space arcs or nodes to their counterparts in the static network.

The concepts of shipment, commodity and net supply are similar to those in the static network, except that all origins and destinations are time-space nodes. We assume that all freight originating from the pickup operation enters the network at 7:00 pm of the shipping day, and that all freight is due at the destination location at 8:00 am of the due day, calculated according to the service level of the shipment. Each commodity is still represented by k, but k is now the time-space destination node. Suppose the set of all commodities in the time-space network is K. The handling

cost is denoted by $h_u = h_i$, where $i = s(u)$. The transportation cost is defined as $c^k_{uv} = c_{ij}$, where $(i, j) = s(u, v)$.

For the load plan, the same notation is adopted as in the previous part. The load plan is represented by binary decision variables $\{y^d_{ij}\}$, where $i, j, d \in V$. The commodity flow is represented by the set of non-negative continuous decision variables $\{x^k_{uv}\}$, where x^k_{uv} is the flow of commodity k on time-space arc $(u, v) \in E_T$, in term of fractional trailers. Finally, τ_{uv} represents the actual number of trailers on (u, v), including loaded and empty ones. The volume of shipments handled at terminal u, in terms of fractional trailers, is denoted by $z_u = \sum_{v \in \delta^+(u)} \sum_{k \in V} x^k_{uv}$.

Thus, the load plan design problem in the time-space network is formulated as below.

$$\text{Minimize} \quad \sum_{(u,v) \in E_T} c_{uv}\tau_{uv} + \sum_{u \in V_T} h_u z_u \tag{14}$$

$$\text{subject to} \quad \sum_{v \in \delta^+(u)} x^k_{uv} - \sum_{v \in \delta^-(u)} x^k_{vu} = b^k_u, \forall u \in V_T, \forall k \in K \tag{15}$$

$$x^k_{uv} \leq M y^{s(k)}_{s(u,v)}, \forall (u, v) \in E_T, \forall k \in K \tag{16}$$

$$\sum_{j \in \delta^+(i)} y^d_{ij} \leq 1, \forall i \in V, \forall d \in V$$

$$\sum_{k \in K} x^k_{uv} \leq \tau_{uv}, \forall (u, v) \in E_T \tag{17}$$

$$\sum_{v \in \delta^+(u)} \tau_{uv} - \sum_{v \in \delta^-(u)} \tau_{vu} = 0, \forall u \in V_T \tag{18}$$

$$z_u = \sum_{v \in \delta^+(u)} \sum_{k \in K} x^k_{uv}, \forall u \in V_T \tag{19}$$

$$x^k_{uv} \geq 0, \forall (u, v) \in E_T, \forall k \in K$$

$$y^d_{ij} \in \{0, 1\}, \forall (i, j) \in E, \forall d \in V$$

$$\tau_{uv} \in Z^+, \forall (u, v) \in E_T$$

Similar to the load plan design model using a static network, the objective function (14) minimizes the total operational cost, including the total transportation cost and

the handling cost. Constraints (15) ensure the mass conservation of each commodity at each time-space node. Constraints (16) ensure the routes of each commodity are consistent with the load plan in the static network. Constraints (17) ensure that there are enough trailers on each arc in the time-space network. Constraints (18) ensure that trailers are balanced through empty trailer repositioning.

It is often more useful to work with path-based formulations. First, we present a path-based formulation to the problem with splittable commodities. Suppose P^k is the set of all feasible paths for commodity k, connecting the origin time-space node feasibly to the destination time-space node. For a path $p \in P^k$, the total handling cost per trailer is h_p. Let x_p^k be the decision variable representing the flow of commodity k on path $p \in P^k$. Then, the path-based formulation of the load plan design problem is presented below.

$$\text{Minimize} \quad \sum_{(u,v)\in E_T} c_{uv}\tau_{uv} + \sum_{k\in K}\sum_{p\in P^k} h_p x_p^k$$

$$\text{subject to} \quad \sum_{p\in P^k} x_p^k = q_k, \forall k \in K$$

$$x_p^k \leq M y_{ij}^{s(k)}, \forall (i,j) \in s(p), \forall p \in P^k, \forall k \in K$$

$$\sum_{j\in\delta^+(i)} y_{ij}^d \leq 1, \forall i \in V, \forall d \in V$$

$$\sum_{\{p\in\cup_{k\in K}P^k:(u,v)\in p\}} x_p^k \leq \tau_{uv}, \forall (u,v) \in E_T$$

$$\sum_{v\in\delta^+(u)} \tau_{uv} - \sum_{v\in\delta^-(u)} \tau_{vu} = 0, \forall u \in V_T$$

$$x_p^k \geq 0, \forall p \in P^k, \forall k \in K$$

$$y_{ij}^k \in \{0,1\}, \forall (i,j) \in E, \forall k \in V$$

$$\tau_{uv} \in Z^+, \forall (u,v) \in E_T$$

Furthermore, if we assume the commodities are non-splittable, then we can create an alternate formulation where the binary variable $x_p^k = 1$ means that commodity k

uses path p; otherwise, it is equal to zero. The resulting formulation is:

$$\text{Minimize} \quad \sum_{(u,v)\in E_T} c_{uv}\tau_{uv} + \sum_{k\in K}\sum_{p\in P^k} h_p q^k x_p^k$$

$$\text{subject to} \quad \sum_{p\in P^k} x_p^k = 1, \forall k \in K$$

$$\sum_{p\in P(k):(u,v)\in p} x_p^k \le y_{s(u,v)}^{s(k)}, \forall (u,v) \in E_T, \forall k \in K$$

$$\sum_{j\in\delta^+(i)} y_{ij}^k \le 1, \forall i \in V, \forall k \in V$$

$$\sum_{\{p\in\cup_{k\in K}P^k:(u,v)\in p\}} q_k x_p^k \le \tau_{uv}, \forall (u,v) \in E_T$$

$$\sum_{v\in\delta^+(u)} \tau_{uv} - \sum_{v\in\delta^-(u)} \tau_{vu} = 0, \forall u \in V_T$$

$$x_p^k \in \{0,1\}, \forall k \in K, \forall p \in P^k$$

$$y_{ij}^k \in \{0,1\}, \forall (i,j) \in E, \forall k \in V$$

$$\tau_{uv} \in Z^+, \forall (u,v) \in E_T$$

3.4 Risk Consequence Assessment via Dispatch Simulation

In this chapter, we focus on the operational risk arising from uncertainty in transit time and the consequence assessment of the risk. Suppose that transit time delays may occur for various timed directs, which correspond to the transportation arcs in the time-space network. If a direct suffers a delay, then all trailers and shipments dispatched on that direct are assumed to be affected.

Suppose in a time-space network $G_T = (V_T, E_T)$, the transit time of each timed direct $(u, v) \in E_T$ is a continuous random variable T_{uv}. Suppose direct $(i, j) \in E$ is the direct corresponding to (u, v) in the static network $G = (V, E)$. Assume that the mean of T_{uv} is t_{ij}, which is the nominal value of transit time on (i, j). We ignore the case of early arrivals, since we assume that early arriving trailers can simply wait to be unloaded at the normal time; it is not difficult to perform a similar analysis if

this is not the case. Then, the delay on (u, v) is represented by a random variable $D_{uv} = (T_{uv} - t_{uv})^+$, where the function $x^+ = \max\{x, 0\}$. The distribution of the delay D_{uv} is completely determined by the distribution of the transit time T_{uv}.

In this research, we will refer to a scenario using a transit time random vector $\mathbf{T}$ realization for all transportation arcs in the time-space network, *i.e.*, $\mathbf{T} = [T_{uv} : (u, v) \in E_T]$. We denote the set of all scenarios by Ω and each individual scenario is denoted by $\omega \in \Omega$. Each ω corresponds to a realization of vector $\mathbf{T}$.

To determine how a load plan will perform in practice, we develop a dispatch simulator in this research. The dispatch simulator evaluates the performance of a given load plan by simulating the trailer dispatches that would result under a specific control policy and a load plan. The simulator thus enables the computation of the total operational cost of the system. When the transit time of each direct is deterministic and given, this total cost is computed by running the dispatch simulation once using the deterministic data. When direct transit times are random, the total cost can be obtained by using a sample average, *i.e.*, generating a reasonably large number of scenarios, running the dispatch simulation for each scenario and computing the average cost over all scenarios. In this research, we assume the transit time of each direct is normally distributed, and independent. More accurate distributions of transit time can be determined by analyzing historical data used in practice.

As an aside, it is important to note again that all trailers that use a specific time-space direct will be assumed to require the same transit time; they will all experience the same delay. It is not any more difficult to assume that individual trailers (or sets of trailers) dispatched on the same direct would experience differing delays. For computational simplicity, we make the simpler assumption.

3.4.1 Control Policy

To the best of our knowledge, most of the research work on the load plan design problem, especially in the time-space network, is based on the assumption that each shipment follows a schedule, usually represented by a timed path. For example, a shipment from CIN to BHM may follow the route CIN-ATL-BHM according to the load plan. The schedule may be as follows: the shipment leaves CIN at 19:00, arrives at ATL at 3:00 on the next day, leaves BHM at 5:00 and arrives at BHM at 8:00 in the morning. However, different from the airline industry in which the flights normally follow a fixed schedule, the LTL trucking industry does not rigidly schedule the dispatch of trucks in practice. Instead, trucks are dispatched according to a set of rules, which we refer to here as a *control policy*. When a truck is delayed, the control policy is still effective, but a pre-determined schedule will be disrupted and may no longer be valid. Thus, we assume that the trailers are loaded and dispatched according to the control policy below:

- **Fixed route:** The physical route (without time) of each shipment is always determined by the load plan and is not changed under any circumstances. For example, if a shipment is delayed, it cannot be expedited by skipping any intermediate terminals.

- **Early cut-time loaded first:** The cut time is the latest time at which a shipment can be dispatched and still arrive at the destination on time. The shipments with early cut time have a priority when loading into a trailer. The other shipments are loaded on a first-come-first-load basis. We assume that the commodity representing all freight bound for a specific time-space destination is infinitely splittable during loading.

- **Dispatch when filled or at earliest cut-time:** A trailer is dispatched when either of the following two conditions is met: (1) The trailer is nearly full, *i.e.*,

the loaded shipments have exceeded a threshold of weight or cube; (2) At least one shipment in the trailer must go, *i.e.*, the earliest cut-time of one of the loaded shipments has been reached.

- **Delay effective:** If a delay occurs, the trailers are still dispatched according to the above rules.

As we can see, the control policy is simple and easy to implement, given a load plan. Furthermore, this control policy is very close to how actual LTL carriers operate their cross-docks. The only key concept that we omit is that dispatches sometimes must wait for driver availability.

3.4.2 Dispatch Simulation Algorithm

We now describe the algorithm that we use to implement the dispatch control policy given a fixed load plan. The static network is represented by $G = (V, E)$, where V is the set of terminals and E is the set of directs. The corresponding time-space network, obtained by expanding G over a discrete time horizon T, is again denoted by $G_T = (V_T, E_T)$, where V_T is the set of time-space nodes. Each time-space node represents a terminal-time pair (i, t), where $i \in V$ and $t \in T$. Suppose now that the time-space nodes are sorted first by the local time and then by the time zone of the terminals, so that the dispatches occur in time order as the nodes are traversed.

For each time-space node $u = (i, t) \in V_T$, a list $\Phi[u]$ is maintained to store the shipments ready for departure from terminal i at time t. In the list, a shipment (d_t, f) is represented by its timed destination, *i.e.*, the time-space node for destination, $d_t = (d, t)$ and the fraction of trailer required f. Initially, each shipment with its timed origin u is added to the list $\Phi[u]$. Note that the time epoch for the origin is 19:00 on the shipping day and that for the destination is 8:00 on the due day, respectively.

The list $\Phi[u]$ for the timed node u is sub-divided by the terminals that the trailers are outbound for next. Each shipment is assigned to the appropriate subdivision

when added to the list. The subdivision is denoted by$\Phi[u][j]$, where the shipments in it are sent to terminal j via direct moves. In each subdivision, the shipments are sorted by cut time in order to efficiently implement the control policy.

The dispatch simulation algorithm proceeds through all time-space nodes at time period t, before proceeding to the first time-space node at the next time epoch. When processing node u, the shipments to be dispatched now are determined. Suppose the shipments outbound to node v fill up to f trailers (f may not be an integer). According to the dispatch rule, at least $\lfloor f \rfloor$ trailers are certainly dispatched. A simple check on the last trailer is sufficient to determine whether it is dispatched at this time; either it is filled to the threshold, or must be dispatched by the earliest cut time rule.

Any shipments that were not dispatched at this time are sorted into the list for the time-space node representing terminal i at the next time epoch. The dispatched shipments are sorted into the list $\Phi[v]$ via transportation arc $(u, v) \in E_T$, where $j = s(v)$. Note that if the transit time of the transportation arc is random, then the time-space node v needs to be determined in real time by the algorithm, according to the randomly generated transit time.

Given the load plan and the set of shipments, the total handling cost can be easily computed, since the load plan determines the route of each shipment and the timing does not affect the handling cost. Thus, it is only the total transportation cost that needs to be computed during the simulation. Suppose τ trailers are dispatched on transportation arc $(u, v) \in E_t$, then the corresponding transportation cost is $c_{uv}\tau$, where c_{uv} is the linearized cost per trailer on that arc.

Suppose that there are n terminals and T time epochs in the time-space network. Also suppose the average number of shipments handled at each time-space node is k, then the complexity of the dispatch simulation algorithm is $O(knT)$, which is proportional to the size of time-space network and the number of shipments. The

pseudo-code of the algorithm is provided in Algorithm 3.

Algorithm 3 Dispatch Simulation Algorithm

Initialize an array Φ of size nT, in which each element $\Phi[u]$ is a list for time-space node $u \in V_T$.
for each shipment (d_t, f) with the timed origin u, add the shipment to $\Phi[u]$.
for each time-space node $u = (i, t) \in V_T$ (sorted) **do**
  for each outbound terminal j in $\Phi[u]$ **do**
    Sort the shipments in $\Phi[u][j]$ by cut time;
    Determine the shipments to be dispatched and those not to be dispatched;
    Determine the number of trailers to be dispatched and the transportation cost;
    Determine the time-space node $v = (j, t')$ to receive the dispatched shipments; (The node is determined by deterministic or randomly generated transit time.)
    The dispatched shipments, except those whose final destination is node j, are added to the appropriate subdivision of list $\Phi[v]$;
    For those arriving at the final destination, check whether they are delivered on-time;
    The un-dispatched shipments are added to the list $\Phi[u']$, where $u' = (i, t')$ and t' is the next time epoch of t.
  end for
end for
Report statistics such as: total number of trailers, total transportation cost, average load factor, delay rate and etc.

Since the dispatch simulation is an accurate assessment of the cost of a load plan given a set of transit times, it is useful to see how accurate the load plan design models are in terms of cost evaluation. We use a test dataset to perform this comparison for the path-based non-splittable load plan model given earlier, modified by removing the empty balance constraints. The test network has 30 terminals, including 8 break-bulks and 22 EOLs. Starting with a given load plan, we use an integer-programming based local search procedure (see [35]) to improve it. During the search, many improving load plans are identified. We randomly select a subset of these load plans. The costs of these plans, as estimated by the load plan design model, are illustrated by the dashed blue line in Figure 13. We can see that these cost estimates are decreasing by definition. Then, we evaluate the selected load plans by using the dispatch simulator. The solid lines illustrate the costs of the load plans under different levels of transit

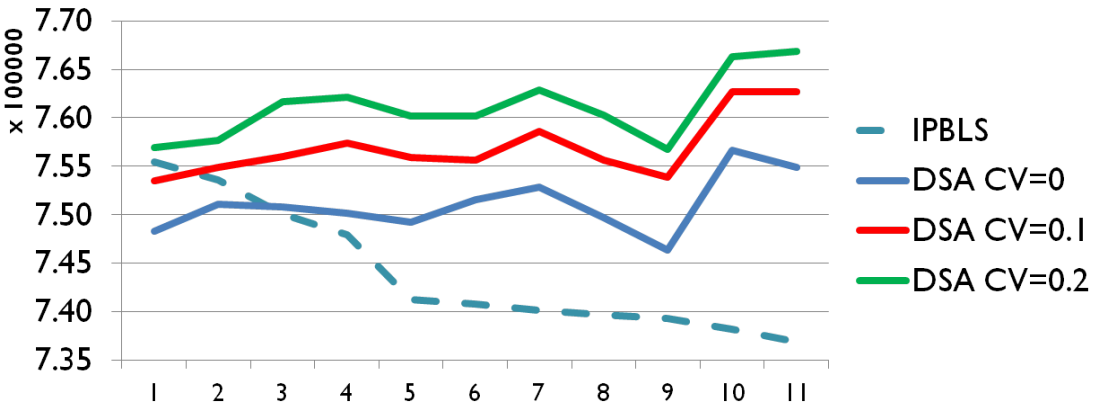

Figure 13: Cost Comparison: Modeled Cost (IPBLS) vs. Dispatch Simulation Cost for a Sequence of Improving Load Plans

time variability, where each direct transit time is drawn from a normal distribution with mean that is the nominal travel time when computing the delay. There is no trend of cost reduction for any of these solid lines. Since the dispatch simulation gives a cost estimate closer to reality, then it seems clear that the local search algorithm may not always be finding actual load plan improvements as it finds plans that appear better to the model. In addition, as the variability of transit time increases, the gap between the model costs and the dispatch simulation costs becomes larger. In the figure, CV stands for the coefficient of variation, which is the ratio of the standard deviation to the mean of a random variable.

3.5 Robust Load Plan Design Algorithm

In this section, we describe a method to obtain a robust load plan that is resistant to uncertainty in the transit time. In theory, the robust version of the load plan design problem can be formulated as a stochastic program based on the formulations described in section 3.3. In practice, it is very difficult to create a useful exact robust model since: (1) the deterministic load plan design problem is already very difficult to solve using exact optimization methods; and (2) there is no simple way to quickly evaluate the change in cost (the recourse cost) when the timed paths are disrupted

by delays.

An alternative approach to building a robust load plan is to rely on a dispatch simulation to compute the costs of a given plan (under uncertainty), and then to build a search heuristic that calls the dispatch simulation to evaluate proposed plans. We propose such a local search heuristic in this section.

3.5.1 Local Search Framework for Problems with Deterministic Transit Times

Before considering the problem with transit time uncertainty, we first note that this type of heuristic scheme may also be an appropriate approach to improve load plans assuming that transit times are deterministic. A local search algorithm starts from an initial solution and then iteratively moves to a better solution, if any, in the neighborhood of the current solution. In the load plan design problem, a solution is a load plan for the given LTL network. The neighborhood of a load plan is defined to be the set of load plans obtained by changing a part of the solution using a limited set of operations. The generic local search framework is described in Algorithm 4.

Algorithm 4 Generic Local Search Algorithm

```
Start with a given load plan y = y_0;
Compute the cost z = f(y) by running the dispatch simulation.
while if there is an improvement within a preset number of iterations do
  for each load plan y' ∈ N(y) do
    Compute the new cost z' = f(y') by running the dispatch simulation;
    if there is an improvement in cost, i.e., z' < z then
      Perform the change: y = y' and z = z'.
    end if
  end for
end while
```

The basic idea of the generic local search algorithm is this: iteratively proposing a local change to the current load plan, evaluating the proposed load plan by dispatch simulation and realizing the change if there is a cost improvement.

To clearly present the idea, we adopt the notation used earlier in Section 3.3. The

load plan is represented by y, which is the solution of the problem. The commodity flow in the time-space network is represented by x. The objective value, *i.e.*, the total cost, is represented by z. Since the number of loaded trailers can be easily obtained given the commodity flows x, then the total cost can be considered as a function of x, which itself can be viewed as a function of the load plan y. Thus, the total cost is represented as a function of the load plan $z = f(y)$. Note that the function $f(y)$ is only a mathematical representation. The actual cost is determined by the dispatch simulation. The generic neighborhood of a load plan y, which is a set of load plans, is represented by $N(y)$. We will discuss the design of the neighborhood in the next section.

3.5.2 Design of Local Search Neighborhood

Under the in-tree assumption, a load plan is comprised of a set of in-trees, one for each destination $d \in V$. In the sequel, we will use the term in-tree and tree interchangeably for simplicity. Suppose the static network is denoted by $G = (V, E)$, where V is the set of terminals and E is the set of directs. Suppose $\mid V \mid = n$, then a load plan can be represented by a set of n trees $P = \{T_d : \forall d \in V\}$, where each element T_d is a tree destined to a terminal $d \in V$. Given the set of commodities K, the flows on tree T_d represent the shipments whose final destination is terminal d. The common directs on different trees provide opportunities for shipment consolidation.

The basic idea of the neighborhood design is to form neighbors that attract additional commodity flow to a single specific direct by changing the structure of multiple trees. Having a higher flow volume on "good" directs may reduce the total cost by enabling better consolidation.

Suppose we seek the possibility of attracting commodity flows onto a direct $(i, j) \in E$. First of all, a feasible tree T_d from which flow might be attracted must not yet contain (i, j) and its destination, *i.e.*, the root of the tree, must not be terminal

i. Thus, a necessary condition on the destination trees that might be modified to contain direct (i, j) for flow routing is that: the feasible trees must be in the set $S_1 = \{T_d : i \neq d \text{ and } (i, j) \notin T_d\}$.

Second, to preserve the structure of the in-tree, the original tree arc $(i, j') \in T_d$ must be removed at the same time of adding (i, j), since every node has one and only one outgoing arc (except the root node, which has none). The process of changing a node's outgoing arc is called a *pivot*.

Third, to form a valid tree, the pivot must not create a directed cycle. We can show that for a tree in S_1, if node i is not on the directed path from node j to the root node d, then, the pivot from (i, j') to (i, j) forms a valid tree. Denote the directed path from j to d by $p(j, d)$, then the pivot is feasible for the trees in the set $S_2 = \{T_d : i \neq d \text{ and } (i, j) \notin T_d \text{ and } i \notin p(j, d)\}$. Thus, the new tree after the pivot is represented as $T'_d = T_d \cup \{(i, j)\} \backslash \{(i, j')\}$.

Fourth and finally, the service level of each origin-destination pair (o-d pair) must be guaranteed after the pivot. If the total transit time from the origin to destination (including handling time) is no more than the maximum transit time allowed by the service level of the o-d pair, then the o-d pair is considered service feasible; if all o-d pairs are service feasible after the pivot, then the pivot is service feasible. Otherwise, the o-d pair is service infeasible and then the pivot is prohibited.

We check the service feasibility by a labeling process. Consider the o-d pairs with a common destination d. On tree T_d, denote the total transit time from node i to the root node d by ϕ_i, which is called the label of node i. By default, the label of d is zero, *i.e.*, $\phi_d = 0$. Then, for each arc $(i, j) \in T_d$, $\phi_i = \phi_j + t_{ij}$, where t_{ij} is the transit time of arc (i, j). If the label of the origin node i is no more than the maximum allowed transit time from i to d, then the service from i to d is feasible. For other destinations, the service feasibility can be checked in a similar way.

Consider the pivot from arc (i, j') to (i, j) on tree T_d. The pivot only affects the

labels of nodes which are on the new sub-tree rooted at node j. Intuitively, they are the upstream nodes of j after the pivot. The label of node j remains the same, and the new label of node i is computed by $\phi_i = \phi_j + t_{ij}$. In addition, the labels of all the upstream nodes change by the same amount. Suppose the original tree T_d is service feasible. If ϕ_i is decreased, then all the upstream labels are decreased, and thus the new tree is certainly service feasible. If ϕ_i is increased, a simple re-labeling procedure starting from node j to the upstream is sufficient to check the service feasibility of the new tree.

For a given arc (i, j), we can use the rules above to determine a set of k feasible trees that could be feasibly modified by adding direct (i, j). The question remains as to the best way to select a subset of those trees to change to find the maximum cost reduction (if any). Theoretically, there are 2^k possible combinations. Even for a small k, it is computationally impractical to check each combination. On the other hand, if we only consider pivots that change a single tree, then the volume of the rerouted shipments may be too small to generate an improvement after the pivot, and the solution may converge to a local optimum. Hence, the algorithm is implemented by using a restricted version of the neighborhood, called *3-tree*, in which the pivots on at most three trees are considered. This turns out to yield a reasonable tradeoff between the efficiency and effectiveness.

For a given direct (i, j), at most three trees in the feasible set are randomly picked, *i.e.*, T_1, T_2, T_3. The pivots are performed on the following sets of trees: $\{T_1\}$, $\{T_1, T_2\}$, $\{T_1, T_2, T_3\}$. For each option, the new cost is evaluated by running the dispatch simulation. If any cost improvement is found, the load plan is immediately updated by the pivots and the 3-tree process terminates for this load plan. If no improvement is found, another 3-tree set is selected, until reaching the maximum number of trials. Algorithm 5 describes the basic 3-tree neighborhood search. A later section will provide some details on enhancements.

Algorithm 5 Basic 3-Tree Neighborhood of a Load Plan

Given a load plan $P = \{T_d, \forall d \in V\}$, initialize the set of candidate directs $D \subseteq E$, to which the commodity flows may be attracted.
for each direct (i, j) randomly picked from set D **do**
  Obtain the set of all feasible trees S;
  while not reaching the preset maximum number of trials **do**
    Randomly pick 3 trees: T_1, T_2, T_3;
    Check the pivot to (i, j) on $\{T_1\}$, $\{T_1, T_2\}$, $\{T_1, T_2, T_3\}$;
    If there is an improvement, do the pivot on the trees and break the while loop.
  end while
end for

Algorithm 6 Basic Robust Local Search Algorithm

Start with a given load plan $y = y_0$;
Compute the nominal cost z of the initial load plan y by running the dispatch simulator on the deterministic transit time;
Compute the sample average cost z_a of the initial load plan y by running the dispatch simulator under N scenarios;
while there is an improvement within a preset number of iterations **do**
  for each load plan $y' \in N(y)$ **do**
    Compute the new nominal cost z' of load plan y;
    If $z' > (1 + \alpha)z$, then discard y' and continue searching;
    Compute the new sample average cost z'_a of load plan y;
    if there is an improvement in the sampling average cost, *i.e.*, $z'_a < z_a$ **then**
      Perform the change: $y = y'$, $z = z'$ and $z_a = z'_a$.
    end if
  end for
end while

3.5.3 Robust Load Plan Design by Sample Average Method

To solve the robust load plan design problem, an algorithm is designed based on the algorithm for the deterministic problem. The key difference lies in the evaluation mechanism of the load plan. The robust algorithm evaluates a given load plan by using a sample-average version of the dispatch simulation, which is described in Algorithm 6. A reasonably large sample of scenarios are drawn from the probability distribution, and the average cost of the load plan over the scenarios is computed by running the dispatch simulation on each scenario. Note that a scenario is represented by the vector of transit time, as mentioned earlier.

Suppose the sample average cost of a given load plan y is represented by $z_a(y)$, in short z_a. In the basic robust algorithm, we use a fixed sample size N. That is, each time a load plan is evaluated, N scenarios are generated and the dispatch simulator computes the cost under each scenario. The performance of the load plan is measured by the average of the above costs.

One main challenge for implementing the robust algorithm is to reduce the time for computing the sample average cost under a large number of scenarios. Suppose it takes a second for the dispatch simulator to compute the cost under one scenario, then the total computation time is N seconds for N scenarios. Suppose there are 1000 solutions in the neighborhood, then the algorithm may need up to $1000N$ seconds, probably measurable in hours, to scan the neighborhood before making an improvement. Thus, we do not check the entire neighborhood to search for a better solution. Instead, a randomized search in the 3-tree neighborhood is used.

Several measures are taken to enhance the efficiency of the algorithm. First of all, the minimum time discretization is used for end-of-line terminals. Each EOL has only two time epochs 19:00 and 8:00 local time each day. This greatly reduces the number of nodes in the time-space network and thus the computation time for the dispatch simulation.

Second, an adaptive sample size is used when evaluating a load plan. The rationale is this: if a load plan has bad performance under the deterministic case and the first few scenarios, then it is likely that we should discard it; on the other hand, if it performs well under the first few scenarios, then we still need to verify its performance under a reasonably large number of scenarios.

Given a candidate load plan y', we first compute its nominal cost z' using deterministic transit time. If z' is too high compared to the current best nominal cost z, *i.e.*, $z' > (1+\alpha)z$, then the candidate load plan is discarded. Here α is a parameter measuring our tolerance for the deterioration in the nominal cost. If $z' < (1+\alpha)z$,

then the average cost of the load plan z'_{a1} is computed under a small sample size, say N_s. If z'_{a1} is greater than the current best average cost z_a, then the load plan is discarded. Otherwise, the load plan is evaluated using more scenarios, say $N - N_s$, where N is a reasonably large sample size. If the average cost over N scenarios z'_{a2} is less than z_a, then the load plan is considered better than the current load plan.

Third, a minimum time horizon is used when generating a time-space network. In order to get an accurate (weekly) cost of a load plan, we run the dispatch simulation on a multi-week time-space network, with repeated weekly shipments. Experiments show that as long as the time horizon is no less than three weeks, the simulation yields an accurate cost. Thus, the 3-week time horizon is used for the dispatch simulation.

3.5.4 Computational Results

The algorithms described above are implemented in C++. The experiments are run on a Lenovo computer with a 2.40 GHz Intel Core (TM) i5 CPU and with 4GB of memory. The algorithm is tested on a 30-terminal LTL network, including 8 break-bulks and 22 EOLs. The dataset is constructed for research purposes based on the service network of a national LTL carrier.

First, we run the deterministic load plan design algorithm on the 30-terminal dataset. The initial solution is a load plan optimized by the IPBLS. The results show that our local search algorithm can provide an additional 6.4% improvement in 3 hours of computation time, when compared to the integer programming-based local search approach given in [35]. Given that IPBLS yields good solutions, our proposed algorithm performs very well. Figure 14 illustrates the improving process of the nominal cost. The horizontal axis represents the ID of each new load plan found during the search.

The algorithm for robust load plan design is similar to the deterministic version. The only difference is that the robust algorithm evaluates each load plan using the

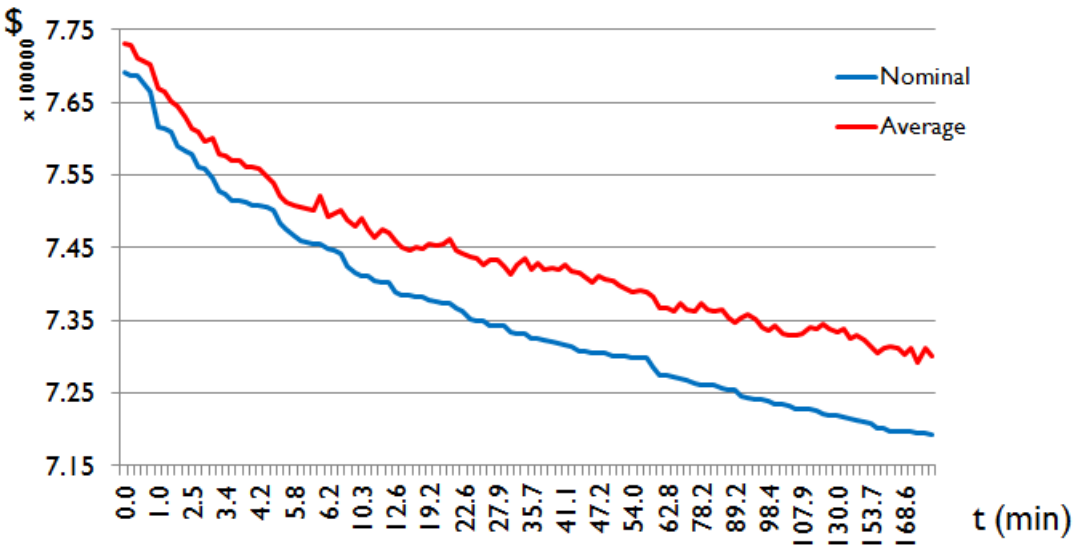

Figure 14: Improvement in Nominal Cost for Test Instance - Deterministic Algorithm

sample average cost, which has been explained in the previous part. For each scenario, the transit time of each arc in the time-space network is randomly generated in real-time, and is assumed to be normally distributed with the mean equal to the expected transit time and the standard deviation proportional to the mean.

During the load plan improvement algorithm, illustrated in Figure 15, the robust algorithm improves the sample average cost while tolerating small deterioration of the nominal cost. In the long run, the robust algorithm can reduce both costs. After an hour, a 3.8% improvement is achieved over the optimal load plan found by IPBLS optimized solution in 4 hours.

3.6 Summary and Future Research

In this chapter, two primary contributions are made. First, a realistic way is provided to evaluate the performance of a given load plan. The dispatch simulation is developed to efficiently compute the cost of a load plan based on a practical control policy. Second, an efficient local search algorithm is proposed to determine a cost-efficient load plan that is also robust under uncertainty in the transit time.

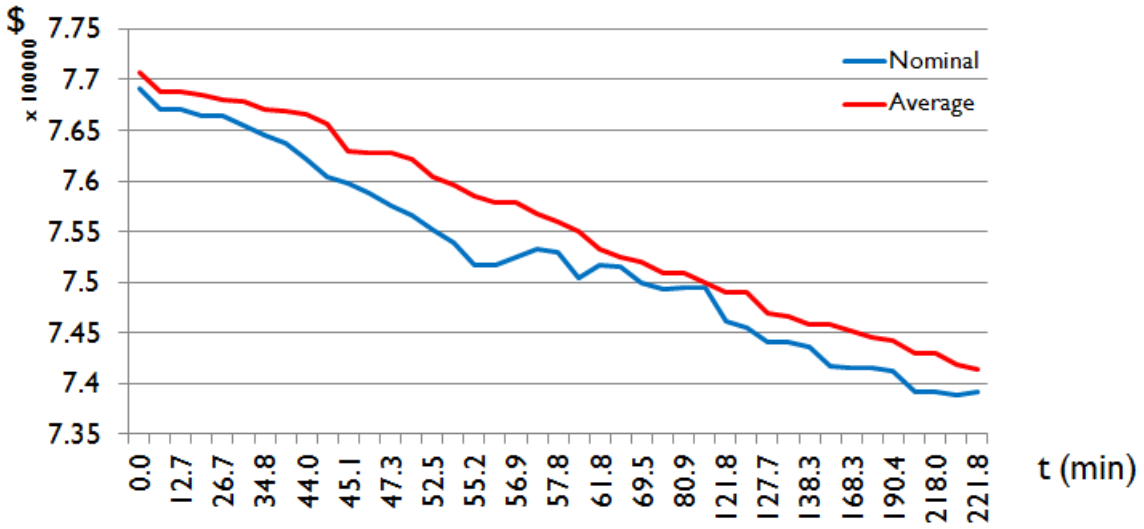

Figure 15: Improvement in Sample Average Cost for Test Instance - Robust Algorithm

The challenge of implementing the robust load plan design algorithm lies in the computational complexity. Even the deterministic load plan problem is very difficult to solve. Moreover, the robust version requires us to evaluate a load plan under a sufficiently large number of scenarios, which adds to the computational burden. In the future, our research will focus on improving the performance of the algorithm. Besides improving the efficiency of evaluation process, another important direction is to increase the chance of improvement during each trial. The following ideas can be tested.

(1) Smart rules can be established to restrict the search in the neighborhood. Then each trial may have a better chance of finding an improvement. For example, for the shipments (*e.g.*, at Salt Lake City) that are destined to a nearby break-bulk (*e.g.*, Denver) are likely to ship direct to the destination.

(2) The "affinity" of trees may imply good combinations of tree that change together. For example, the trees destined to Dallas and Houston are likely, but not always, to change together, since the shipments destined to these two terminals may

have similar routes.

(3) High volume directs generally have a high chance to attract flows, while the low volume directs is less likely to attract flows.

(4) If the neighborhood is too restricted, the algorithm may quickly end up with a local optimum. In addition, the number of directs in set D only decreases in the current setting. We may have a large set of candidate direct for attracting flow. One way is to add long directs (by skipping the intermediate break-bulks) close to the destination.

CHAPTER IV

DEFENDING A FOOD SUPPLY CHAIN AGAINST INTENTIONAL ATTACK

4.1 Introduction

Fast and efficient food supply chains may be used as a platform by an intelligent adversary, *i.e.*, a terrorist group, to deliver a chemical or biological toxin to a population to cause excessive morbidity and mortality. The objective of our research is to present a methodology to assess and mitigate the risk of an intentional attack to a food supply chain.

In this introductory section, we provide a conceptual framework for assessing and mitigating the risk of intentional attacks to a general food supply chain. First of all, the agents which can be used for attack need to be identified. The list of possible agents for attacks can help the defender prevent, detect and respond to a contamination event. From the attacker's perspective, the ideal agent should easily and widely disseminate in the targeted food supply chain; otherwise, the attacker needs to contaminate a large quantity of food units in order to cause a large impact.

Second, the agent dissemination process in a supply chain needs be studied. Technically, a process model needs to be developed in order to describe the agent dissemination process in a food supply chain. In the next section, a state-space model based on a system of differential or difference equations will be presented. The metric *concentration* is used to quantify the amount of agent in the food products.

Third, a risk consequence assessment needs to be performed based on the system model. The consequence can be measured by mortality and morbidity, *i.e.*, the number of people dead and sick because of the attack. There are also other metrics of the

consequence. For the attack, an intentional contamination is carried out by inserting the agent into a supply chain component. Thus, an attack can be represented by the mass of agent inserted, the insertion point and the time of insertion. To assess the consequence of attack, either a simulation is run based on the system model, or a functional relationship is established between the consequence and the parameters of the attack.

Finally, a risk mitigation strategy needs to be developed to defend the supply chain against a potential attack. One countermeasure is to develop a defensive strategy and protect the supply chain with external defensive resources. Another way is to mitigate the risk of an intentional attack with a better design and control of the supply chain.

The main topic of each section in the remainder of this chapter is now summarized. In section 4.2, the related literature is reviewed. In section 4.3, a modeling scheme is provided to describe the agent dissemination process in a general food supply chain. The process is represented by a state-space model, which can be used to support further analyses. In section 4.4, a case study is presented using a simple liquid egg supply chain, in order to illustrate our risk assessment methodology. First, a process model is built based on industrial data. Then, the consequence assessment is performed by running a simulation based on the system model. Moreover, an in-depth analysis is performed to determine the worst-case consequence given a smart attack, which considers the operational characteristics of the system, mainly phases of system components. The worst-case consequence assessment algorithm proposed efficiently estimates the maximum consequence of attack, and will be used for further analyses on the supply chain.

4.2 Related Literature

Recently, researchers are paying more and more attention to food security with respect to intentional contamination. [16] is considered the first paper that attempts

to comprehensively and systematically examine all incidents of intentional and malicious contamination of foodstuffs along the entire food supply chain. Many examples of food incidents are given. From the examples, several observations are made: pre-harvest contamination of animal and plant production is rare; and post-harvest contamination (product assembly, processing, packaging, and storage) is more frequent than pre-harvest, but is still somewhat rare due to the security control of the processing facility. Most contamination incidents occur at retail, food service and end consumers. Most events resulting in death and injuries also occurs from these types of incidents. [37] also gives many examples of deliberate contamination in the food supply chain. [57] describes the threat of a biological terrorist attack from the public health perspective. Possible biological agents and their characteristics are provided. Methods for detection and surveillance are briefly introduced.

In recent years, products have been increasingly distributed from central facilities, a development that has markedly increased the risk of large outbreaks. In 1994, a large common-vehicle outbreak in the United States affected $224,000$ individuals in 41 states with Salmonella enteritidis due to an ice cream pre-mix contaminated in a liquid tanker truck which had previously transported liquid, unpasteurized eggs (see [42]). Although production facilities have not been successfully targeted to date, a successful attack on such a facility could lead to large consequences and it is important to be able to assess the potential consequences.

Quantitative consequence assessment of a food system attack has received less attention. One exception is [65], which studies the consequence if a milk supply chain were successfully attacked by botulinum toxin. In the paper, the dynamics of the milk supply chain is described by a system of linear first-order differential equations. Then, the dissemination of agent given a fixed amount of botulinum toxin released to raw milk silos is analyzed largely with queuing theory and reliance on Little's Law. The paper assumes the agent can only be introduced at milk farms or tanker

trucks. Both cases are equivalent to contaminating the raw milk silos. The operating cycles of the raw milk silos are divided into three periods: filling, replenishing and draining. The concentrations of the agent in different tanks are computed. The resultant concentration is the average of the instant concentration in the input flow over a time period. The effect of pasteurization is also studied. The percentage of deactivated toxin is inferred based on data of similar food under similar processing conditions. Finally, the number of people affected by the contaminated gallons of milk is computed. In addition, the probability of poisoning is computed based on the dose-response curve governed by a probit model.

Although some ideas in our study are similar to those in [65], our consequence assessment method can provide the consequence of attack under different designs of the supply chain. In addition, we provide in-depth analyses at the operational level of the supply chain processes, which will be useful to support the improvement of the supply chain design and control in order to reduce the risk of being attacked while maintaining a satisfactory level of productivity.

4.3 Methodology for a General Food Supply Chain

In this section, a risk assessment and mitigation methodology is provided to address an intentional attack to a general food supply chain, according to the conceptual framework described in Section 4.1. First, relevant measures and notations are given. Second, a general modeling scheme is provided to describe the dissemination of a contaminating agent in the supply chain. Although the dissemination of the agent is different in solid food and in liquid food, both cases can be abstracted by a state-space model. Finally, further analysis can be performed based on the system model for the purpose of risk assessment and mitigation.

4.3.1 Notations and Assumptions

In our research, we consider a general toxin agent. Suppose the agent is inserted into a certain component of a targeted food supply chain. Then, the attack can be characterized by the mass of the agent m, the insertion point i, and the time of insertion t. Thus, a triple (m, i, t) can be used to represent an attack.

In this chapter, we define time zero to be the time when the agent is inserted into the system. That is, the time of attack is defined to be $t = 0$. Suppose the mass of agent in a container is denoted by $m(t)$, then the initial mass of agent in that container is $m(0)$.

The ultimate goal of the attack is to cause morbidity and mortality, *i.e.*, the number of people sick and dead, by contaminating the food delivered by the supply chain. To simplify the analysis here, we measure the consequence of an attack by the number of contaminated food units (*e.g.*, packages) in which the concentration of the agent is above a threshold level. With field knowledge, experts in public health can transform our results to morbidity and mortality figures. Furthermore, it is not very difficult to use our approaches to instead determine a distribution of packages with various levels of toxin concentration as well.

In this chapter, we focus on liquid food which is widely regarded as a major contamination risk. Like others, [65] has identified botulinum released in liquid food as one of the three greatest threats to humans, together with a smallpox attack and an airborne anthrax attack. Generally, toxin is easiest to disseminate within liquid or extruded food products and more difficult to disseminate within solid foods. However, our methods may be adapted to assess solid food supply chains. For liquid food, *e.g.*, milk and juice, the concentration of the agent in the food product at any given time is given by $c(t) = m(t)/V(t)$, where $V(t)$ is the volume of liquid food in the container.

Consider a liquid food supply chain. Suppose the velocity of liquid flowing out of a container or process is δ (measured for example in units m^3/s), then the time

needed to drain the container is $T_d = V/\delta$, where V is the (maximum) volume of liquid in the container. If the container is assumed to have a constant liquid level, then the inflow velocity is equal to outflow velocity, and thus, the time needed to fill the container is also T_d.

The dissemination of agent in the system is influenced by three factors: diffusion, convection and liquid flow. In this chapter, we focus on the effect of liquid flow, since the significance of the other two factors is comparatively small. In addition, the following assumptions are made: (1) the concentration of agent in the container is uniformly distributed soon after it is inserted into the container, *i.e.*, $c(t) = m(t)/V(t)$; and (2) the rate that the agent flows out of a container is proportional to its concentration in the container, i.e. $dm(t) = c(t)dV = c(t)\delta dt$.

4.3.2 State-space Model for a Component

Consider the dissemination process of a contaminating agent in a general food supply chain. A system model needs to be developed to describe the input-output (I/O) relationship between the attack and its consequence.

First of all, a model needs to be built for each component of the supply chain system. For each component, a state-space model is easy to build. With a state-space model, the I/O model of the system can be easily determined. Moreover, a simulation model can be easily developed from a set of connected state-space models.

We define the state of a system to be that collection of variables necessary to describe a system at a particular time. If all the possible states of a system are countable, then the states are discrete; otherwise, the system has continuous states. In this chapter, we focus on systems with continuous states.

A continuous-time system is one for which the state variables change continuously with respect to time; a discrete-time system is one for which the state variables change instantaneously at separated points in time. These two concepts are different from

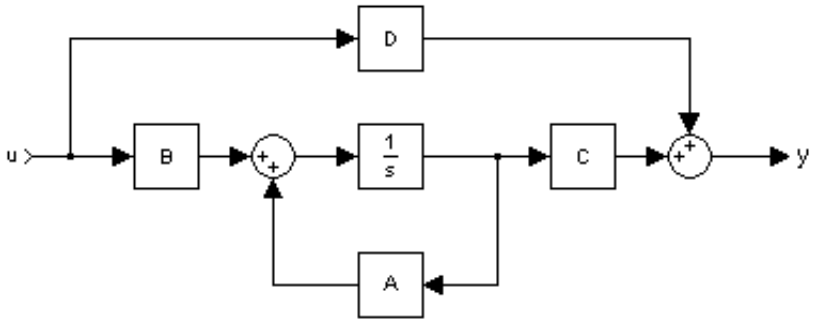

Figure 16: Typical State Space Model

the two in the last paragraph. For simplicity, we illustrate the state-space model using a continuous-time linear model first. Later, the model describing a food system using general functions will be presented.

Consider the following simple state-space model:

$$x'(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

where $x(t)$ is the vector representing the state of the system; $x'(t)$ is the derivative of $x(t)$; $y(t)$ is vector representing the outputs of the system; and $u(t)$ is the vector representing the inputs of the system. Suppose there are p inputs, q outputs and n state variables in the system, *i.e.*, $x(t) \in R^n$, $u(t) \in R^p$ and $y(t) \in R^q$, then, the matrices have the following dimensions: $A \in R^{n \times n}$, $B \in R^{n \times p}$, $C \in R^{q \times n}$, $D \in R^{q \times p}$.

In this state-space model, the first set of equations represents the dynamics of the system, and the second set of equations represent the outputs of the system. The state-space model can be illustrated with a system diagram, provided in Figure 16. A simulation model can be obtained easily from the diagram. Note that for a continuous-time system, the system dynamics are characterized by differential equations; for a discrete-time system, difference equations are used in the state-space model.

To model the agent dissemination process in a system component, *e.g.*, a container, the state variable is the mass of agent $m(t)$ in the container; the output variable is

the concentration of the agent $c(t)$ in the outflow; the input variable $u(t)$ is the concentration of the agent in the inflow; the attack is represented by the initial condition of the dynamic equations. Note that the system has a single state variable, a single input and a single output. Given the process is continuous in time, the general state-space model is illustrated as follows:

$$\begin{aligned} m'(t) &= F_t(m(t), u(t)) \\ c(t) &= G_t(m(t), u(t)) \\ m(0) &= m_0 \end{aligned}$$

where $F_t(m, u)$ and $G_t(m, u)$ are general functions, which may be nonlinear and time-variant. Usually, $G_t(m, u)$ can be simplified as $c(t) = m(t)/V(t)$. In the initial condition, m_0 is the initial mass of agent inserted into the components.

A dynamic equation in the form of $c(t) = f(m'(t), m(t), m(0), u(t))$ can be derived from the state-space model, in order to describe the I/O relationship. And the dynamic equation of the whole system can be determined by composing the equations of each component.

4.3.3 System Model of Supply Chain

Given the state-space model of each component in a supply chain network, we need to develop a model for the whole supply chain by establishing the relationships among the inputs and outputs of the related components. The fundamental law of mass conservation is applied on both the food product and the agent.

First, we study three basic relationship of supply chain components (or supply chain designs): sequence, merge, and distribution, illustrated in Figure 17.

In the *sequence* supply chain, the components are connected in series. Obviously, the concentration of agent in the outflow from the previous component is equal to that in the inflow to the next component, *i.e.*, $c_1(t) = u_2(t)$.

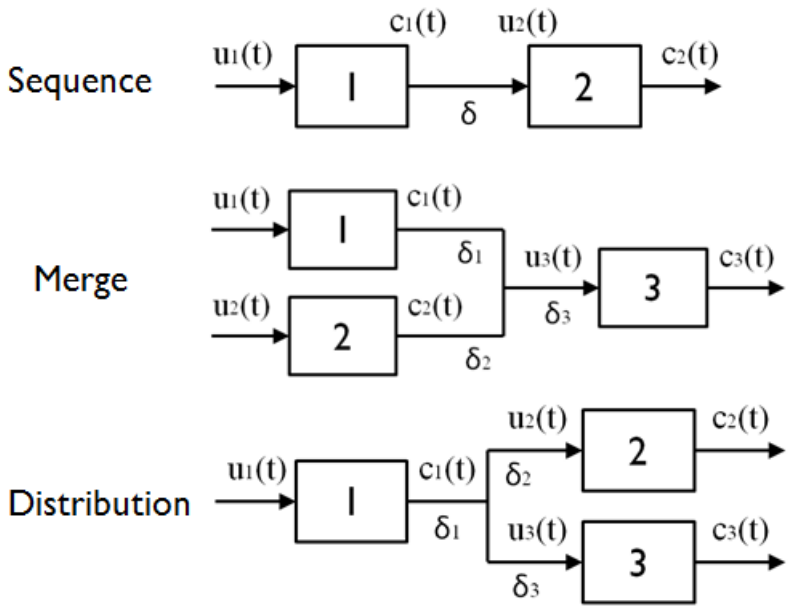

Figure 17: Basic Supply Chain Designs

In the *merge* supply chain, the flows from upstream components merge into a single downstream component. Without loss of generality, we explain with an example of three components in Figure 17. Suppose the flow velocity in the three components are $\delta_1(t)$, $\delta_2(t)$ and $\delta_3(t)$, respectively. Note that they may change over time. Also suppose the concentration of agent in the outflow from component 1 and 2 are $c_1(t)$ and $c_2(t)$, respectively; the concentration of agent in the inflow of component 3 are $u_3(t)$. Then, according to the mass conservation of the food product and the agent, we have:

$$\delta_1(t)c_1(t) + \delta_2(t)c_2(t) = \delta_3(t)u_3(t)$$

$$\delta_1(t) + \delta_2(t) = \delta_3(t)$$

Thus, let $\lambda(t) = \frac{\delta_1(t)}{\delta_1(t)+\delta_2(t)}$, we have

$$u_3(t) = \lambda(t)c_1(t) + (1 - \lambda(t))c_2(t)$$

In the *distribution* supply chain, the flow from a single upstream component is distributed among several downstream components. Without loss of generality, we

explain the ideas with an example of three components in Figure 17. Suppose the flow velocity in the three components are $\delta_1(t)$, $\delta_2(t)$ and $\delta_3(t)$, respectively. Also suppose the concentration of agent in the outflow from component 1 is $c_1(t)$; the concentration of agent in the inflow to component 2 and 3 are $u_2(t)$ and $u_3(t)$, respectively. Thus, according to the mass conservation of the food product and the agent, we have:

$$\delta_1(t)c_1(t) = \delta_2(t)u_2(t) + \delta_3(t)u_3(t)$$

$$\delta_1(t) = \delta_2(t) + \delta_3(t)$$

However, the above equations can only be used to verify the correctness of the concentrations, but they can not yield anything. To obtain the input of downstream components $u_2(t)$ and $u_3(t)$, more information is needed. We discuss two cases of the following setting. Consider three components connected by pipes, which branch after component 1, illustrated in Figure 17. Assume a valve is installed on the pipe before component 2, to control the inflow, and another valve is installed before component 3. For simplicity, the valves can only have two states: open and closed.

Case 1: Suppose two valves are open at all time. Then, component 2 and 3 receive flow proportionally from component 1. Then, $u_2(t) = c_1(t)$ and $u_3(t) = c_1(t)$, for all $t > 0$.

Case 2: Suppose the two valves are open alternatively for 5 time units. That is, the valve on component 2 is open when $t \in T_1$, while the valve on component 3 is open when $t \in T_2$, where $T_1 = [0, 5) \cup [10, 15) \cup [20, 25) \cup \ldots$ and $T_2 = [5, 10) \cup [15, 20) \cup [25, 30) \cup \ldots$. Thus,

$$u_2(t) = \begin{cases} c_1(t) & \text{if } t \in T_1 \\ 0 & \text{if } t \in T_2 \end{cases}$$

$$u_3(t) = \begin{cases} 0 & \text{if } t \in T_1 \\ c_1(t) & \text{if } t \in T_2 \end{cases}$$

A small conclusion for the distribution supply chain: the inputs of downstream components are closely related to the specifics of the operations and have no unique representations. Case 1 above represents a continuous distribution of food product while Case 2 represents a distribution in batches. Thus, we use two general functions f_2 and f_3 to represent them:

$$u_2(t) = f_2(c_1(t), \delta_1(t), \delta_2(t), \delta_3(t))$$

$$u_3(t) = f_3(c_1(t), \delta_1(t), \delta_2(t), \delta_3(t))$$

With the three basic relationships of supply chain components, we can build models for complex supply chain networks. For example, a supplier may serve multiple customers, while each customer is served by multiple suppliers. This multiple-to-multiple relationship can be broken into several basic elements: for each supplier, the outflow can be represented by a distribution relationship; for each customer, the inflow can be represented by a merge relationship. Having the state-space equations and the inter-components equations, we can develop a system model for a supply chain.

On the basis of the system model, risk consequence assessment can be performed by simulation or mathematical analysis. In the following section, we present an example risk assessment using a specific supply chain *i.e.*, liquid egg supply chain.

4.4 Case Study: Liquid Egg Supply Chain

In this section, a case study of the liquid egg supply chain is presented as a proof-of-concept of our methodology. There are several reasons for choosing the liquid egg supply chain: (1) the liquid egg is widely used by food service providers and food manufacturers as ingredient of food product; (2) agent can easily disseminate during the production processes of liquid product and has a large potential to cause a massive public health incident; (3) milk supply chains are more extensively studied while the

more complicated liquid egg system is seldom studied; and (4) the method we develop for the liquid egg system can be generalized to apply to other food systems.

4.4.1 Introduction to Liquid Egg Supply Chain

To better present the ideas, an introduction is given to the egg industry. The US produces 75 billion eggs each year, accounting for 10% of the world's total production. 57.8% eggs directly go to the retailers and are consumed by consumers as shell eggs. 30.8% are further processed and used by food service operators. Those processed egg products, in the form of liquid or powder, are mainly used by food service providers, such as bakers, restaurant chain etc. For those users, liquid egg can reduce the operational cost. Liquid egg may be utilized as ingredient during production, which can greatly simplify the manufacturing process. However, since liquid egg loses its natural protective shell compared to shell egg, it is may be contaminated accidentally in production or intentionally by terrorists.

Nowadays, egg production and processing are often vertically integrated in one facility, including a hen house, an egg processing facility, and transportation vehicles. Figure 18 gives an overview of the egg production and processing procedure. After eggs are laid by the hens, they are transported to the sanitizing and grading facility, as shown in the left part of the figure. After grading, good eggs are packed according to grade AA, A or B, while off-graded eggs are transported to the egg processing facility where the eggs are broken to produce various egg products, *e.g.*, liquid whole egg, liquid egg yolk, liquid egg white, whole egg powder, egg yolk powder, egg white powder, and etc.

The off-graded eggs are stored in trays, a shallow platform designed for carrying eggs, and then transported to the processing facility. In the processing plant, the eggs are loaded onto the processing line from a tray by an automatic egg loader. For example, a typical egg loader has a capacity of loading 108000 eggs/hour. Loaders of

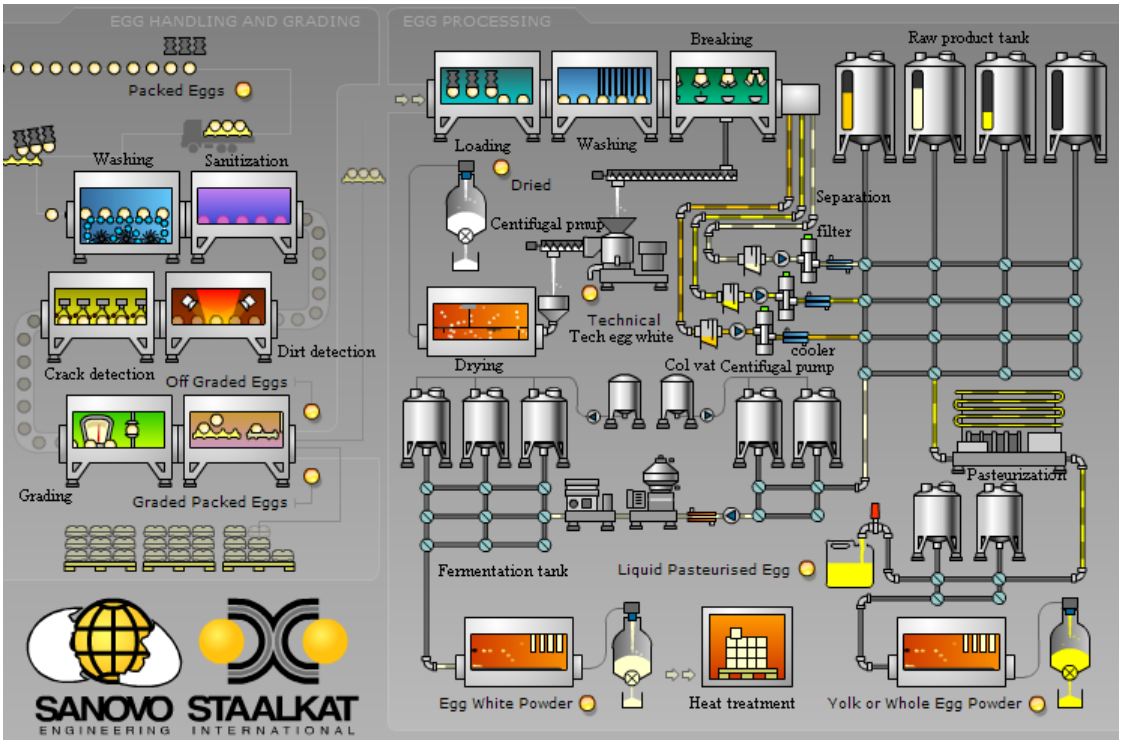

Figure 18: The Egg Processing System

older generation can load eggs at a rate of 54000 eggs/hour.

Before breaking, the eggs are washed for hygienic reasons. The egg is protected by the shell and is naturally sterile. It is not until it is broken that there is a potential risk of contamination by bacteria. Therefore, it is important that hen droppings and other foreign bodies are washed off before the egg is broken. USA authorities require washing times larger than 40 seconds. A typical egg washer can wash 10800 eggs/hour.

After washing, the eggs proceed to the breaking process. The eggs enter the breaker in a line and are processed continuously. There is no batch in breaking. A typical egg breaker can break 5400-32400 eggs/hour. The advanced type has a capacity of breaking 54000-162000 eggs/hour. There are two possible configurations after breaking. One configuration produces liquid whole egg, while the other separates yolk and white and produces liquid egg yolk and egg white. There may be up to three

pipes outgoing from a breaker.

The liquid egg products are collected by collecting vats. The typical capacity of a collecting vat is 130 liters, which makes it possible to connect the vats to every size of egg processing plants. The half cover design facilitates the inner inspection, but leaves a hole for contamination. After the collecting vat, the liquid is pumped by a centrifugal pump and goes through a filter. After cooling (through cooler), the liquid is stored temporarily in raw product tanks. The raw product tank system is used for storing the egg liquid, standardization (mixing of different egg products to obtain a certain egg quality), blending (adding salt, sugar, preservatives, etc), pH-adjustment and fermentation (removing glucose from the egg white prior to spray drying).

Pasteurization is a process of heating liquid egg up to 62-64°C (143.6-147.2°F) for 15 seconds, and then cooling it immediately. Unlike sterilization, pasteurization is not intended to kill all micro-organisms in the liquid egg. It only reduces the number of viable pathogens. But the number will grow again as time goes. After pasteurization, the liquid egg flows into the finished product tank for temporary storage before being filled into packages for distribution.

4.4.2 System Model for Liquid Egg Production System

The actual liquid egg production system is very complex, however, the key components can be reduced to a simplified version, as illustrated in Figure 19. The processes before the breaking of eggs are ignored, since eggs are protected by natural shells before breaking. For simplicity, assume the system produces only one type of product: the liquid (whole) egg. In Figure 19, there are four components in the simplified system: collecting vat, raw product tank, pasteurizer and finished product tank. The system starts from the collecting vat, a small container to collect the liquid egg outcoming from the breaker, as a buffer. The liquid egg continuously flows through the collecting vat and then into the raw product tanks after filtering. In fact, there are

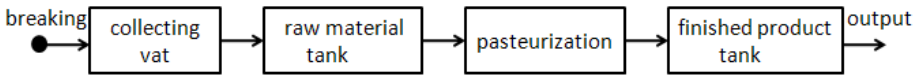

Figure 19: Simplified Liquid Egg Production System

typically a small number of raw product tanks that are used alternately, which will be explained in detail later. In the raw product tank, the liquid egg is blended with necessary ingredients. After being processed, the raw product is drained from raw product tank and pasteurized by flowing through a pasteurizer, which reduces the population of micro-organisms in the liquid egg. Note that some toxin agents are immune from pasteurization. Then, as the finished product, the liquid egg flows into one of the finished product tanks, also used alternately. Finally, the finished liquid egg is filled into final packages for distribution. Although there are multiple package sizes in practice, for simplicity we assume there is only one package size. The reasons to use multiple storage tanks alternately are the following: (1) for qualify control purposes, the liquid egg are processed in batches. In the tanks, the liquid egg may be blended with necessary ingredients; (2) for hygienic reasons, a tank needs to be cleaned after each batch of processing. During this period, another tank needs to take the place and receive the liquid egg from upstream.

Now, we adopt the notations and assumptions in Section 4.3.1, and build models for the components in the liquid egg production system.

4.4.2.1 Collecting Vat

The collecting vat is a small system component (typically 130 liters), which is used to temporarily store the liquid egg. Suppose the liquid level in the collecting vat is controlled to be constant, represented by constant V. It implies that the inflow is equal to outflow, denoted by δ. Suppose that the concentration of agent in the inflow is $u(t)$, then the mass of agent in the vat can be described by the following

differential equation: $m'(t) = \delta u(t) - \delta m(t)/V$. On the right side of equation, the first term represents the rate of agent flowing into the vat and the second term represents that flowing out.

For a collecting vat, the state is characterized by the mass of agent $m(t)$; the input is concentration of agent in the inflow $u(t)$; the output is concentration in the outflow $c(t)$. Suppose the mass of agent inserted into the vat is m_0, which is the initial condition of the system. Thus, the state-space model is described below:

$$\begin{aligned} m'(t) &= \delta u(t) - \delta m(t)/V \\ c(t) &= m(t)/V \\ m(0) &= m_0 \end{aligned}$$

The state-space model can be solved via Laplace Transform. Please refer to appendix for more details about Laplace Transforms. The transform yields:

$$C(s) = \frac{1}{T_d s + 1} U(s) + \frac{1}{s + T_d^{-1}} m_0/V$$

The output in the time domain can be obtained via inverse Laplace transform. The inverse Laplace transform of the first term is the system response for input $u(t)$ and that of the second term is $\frac{m_0}{V} e^{-t/T_d}$, indicating that under zero-input the concentration decreases exponentially to zero, starting from the initial concentration. For example, if the input is a unit step function, whose value jumps from 0 to 1 at time 0, then $U(s) = \frac{1}{s}$, and thus $c(t) = (1 - e^{-t/T_d}) + \frac{m_0}{V} e^{-t/T_d}$.

If the initial concentration is zero, then the collecting vat can be characterized by the typical transfer function, which is called the *inertial component* in control theory:

$$H(s) = \frac{C(s)}{U(s)} = \frac{1}{T_d s + 1}$$

4.4.2.2 Storage Tanks

During the production process, the liquid egg needs to be temporarily stored in tanks. There are two types of storage tanks: raw product tank and finished product tank. In a raw product tank, the raw liquid egg is blended with ingredients before pasteurization. The tank is not drained until the liquid egg is well blended. Different from the raw product tank, the finished product tank serves only as a container temporarily storing the product and thus is similar to the collecting vat when it is full. However, the tanks are filled and drained in an operating cycle. The liquid egg produced during an operating cycle will be called a batch.

In general, the operation cycle of any tank can be divided into four phases: (1) filling (F), (2) blending (B) or replenishing (C), (3) draining (D), and (4) cleaning. During the filling phase, the liquid egg flows into a clean empty tank until the liquid reaches a certain level. The tank has no outflow during this phase. The next phase has two cases: blending or replenishing. During a blending phase, raw product is mixed with ingredients and there is no inflow or outflow; during a replenishing phase, the tank behaves like a collecting vat (C), with constant volume and inflow equal to outflow. The blending phase is represented by the capital letter B and the replenishing phase is represented by the capital letter C. After the second phase (B or C), the liquid in the tank is drained (D) and then the tank is cleaned and sanitized. When a tank is in a draining or cleaning phase, another tank takes its place to receive incoming flow. Thus, the tanks are used alternately.

We assume that raw product tanks have F-B-D phases and finished product tanks have F-C-D phases. In either case, the volume of liquid in the tank (in one operation cycle) can be represented by the following equation, illustrated in Figure 20:

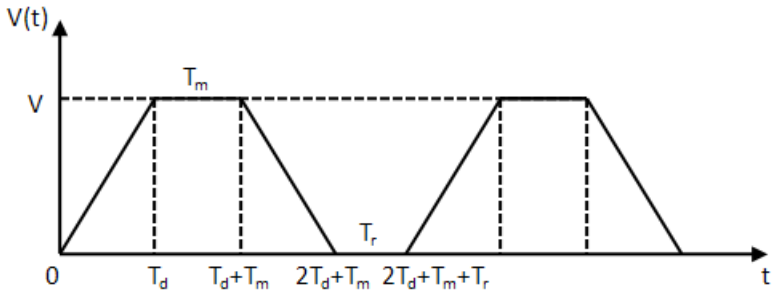

Figure 20: Volume of Liquid in Storage Tank

$$V(t) = \begin{cases} \delta t & 0 \le t < T_d \\ V & T_d \le t < T_d + T_m \\ \delta(2T_d + T_m - t) & T_d + T_m \le t < 2T_d + T_m \\ 0 & 2T_d + T_m \le t \le T_d + T_m + T_r \end{cases}$$

In this expression, we assume that the duration of the filling phase (F), the middle phase (B or C), the draining phase, and the cleaning phase last respectively T_d, T_m, T_d, T_r. The effective duration of an operating cycle is $T_c = 2T_d + T_m$; this is the time during which the tank is filled (partially) with food product. In the sequel, the cleaning phase is usually ignored, since it is unlikely for a tank to be an attack target during this phase.

The basic dynamic equations of a storage tank, if represented in a general form, are similar to those of a collecting vat. But there are two key differences: first, the volume of liquid in the tank changes over time, represented by $V(t)$, while a collecting vat has a constant volume V; and second, not all phases have inflow and not all phases have outflow, while collecting vat has a constant inflow and outflow.

To obtain a general model for the concentration dynamics of a storage tank, we introduce two binary enabling signal functions that indicate whether inflow and/or outflow exist at a given time. Let $e_1(t)$ and $e_2(t)$ signal input and output respectively,

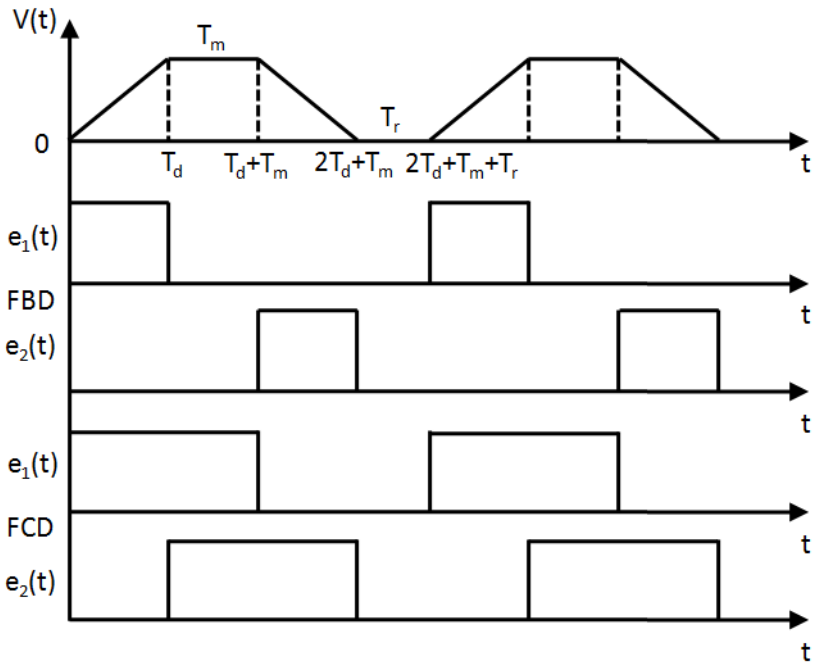

Figure 21: Enabling Signal Functions for F-B-D and F-C-D Storage Tanks

as illustrated in Figure 21. Then, the state-space model of a storage tank is:

$$m'(t) = \delta u(t) e_1(t) - \frac{\delta}{V(t)} m(t) e_2(t)$$
$$c(t) = \frac{m(t)}{V(t)} e_2(t)$$
$$m(0) = m_0$$

The general model can be transformed into a simulation model. However, the model is not linear and thus cannot be solved directly. Instead, we can analyze the model phase by phase separately, as follows.

(1) Filling phase

There is no outflow from a storage tank in a filling phase. Thus, we can simplify the mass dynamics to be $m'(t) = \delta u(t)$. Although the tank is initially clean, a contaminating agent can be inserted during filling phase. Let the time that agent is inserted be $t = 0$, then $m(0) = m_0$. Note that the operating cycle for the tank may

not begin at time 0. Suppose that the operating cycle (which begins with an empty tank filling) starts at time $-t_0$, where $0 \leq t_0 < T_c$. We now define t_0 as the *initial phase* (or offset) of the tank. Then, the volume of liquid in the tank at time $t \geq 0$ is $V(t) = \delta(t + t_0)$. Of course, the tank is filled completely once T_d filling time has elapsed, so this filling phase ends at $t = T_d - t_0$. Given these observations, it is clear then that $m(t) = \delta \int_0^t u(\tau)d\tau + m_0$ and

$$c(t) = \frac{1}{t}\int_0^t u(\tau)d\tau + \frac{m_0}{\delta(t + t_0)}$$

where t_0 is initial phase. As we can see, $c(t)$ is the average of input concentration over time, if the initial condition is ignored.

(2) Blending phase

During the blending phase, there is no inflow or outflow and the volume of liquid is constant. If no agent is inserted during this phase, then the concentration remains unchanged. If agent with mass m_0 is inserted at any time during this phase and the accumulated mass and concentration from the prior phase is denoted as $m(T_d)$ and $c(T_d)$, respectively, then, $m(t) = m(T_d) + m_0$ and

$$c(t) = c(T_d) + \frac{m_0}{V}$$

(3) Replenishing phase

During a replenishing phase, the volume of liquid food in the storage tank remains constant. The tank during this phase is equivalent to a collecting vat, and thus,

$$C(s) = \frac{1}{T_d s + 1}U(s) + \frac{1}{s + T_d^{-1}}c_{T_d}$$
$$c(t) = \mathcal{L}^{-1}\{C(s)\}$$

(4) Draining phase

During a draining phase, there is no inflow and thus,

$$m'(t) = -\frac{\delta}{V(t)}m(t)$$
$$c(t) = \frac{m(t)}{V(t)}$$
$$m(0) = m_0$$

If there is no agent inserted during this phase, then the concentration of agent is constant since before draining the liquid is well blended and the concentration is uniform in the tank. Suppose the accumulated concentration from the prior phase is $c(T_d + T_m)$ and that the initial phase is such that $T_d + T_m \leq t_0 < T_c$. Then, the volume of liquid remaining in the tank when the insertion occurs is $\delta(T_c - t_0)$ and therefore the concentration in a draining tank is constant at

$$c(t) = c(T_d + T_m) + \frac{m_0}{\delta(T_c - t_0)}$$

until the cycle ends at $t = T_c$.

4.4.2.3 Pasteurizer

Assume that the pasteurizer reduces the input concentration of agent proportionately. Suppose the pasteurization factor is $\alpha \in [0, 1]$, then the output concentration of agent can be represented as $c(t) = \alpha u(t)$ and the transfer function is $H(s) = \alpha$.

4.4.2.4 Packaging

Packaging is equivalent to a filling phase of a storage tank. Suppose the volume of the package is V and the filling cycle time is $T_s = V/\delta$, where δ is the filling rate. Since the attacker has no incentive to contaminate a single package, and if we assume that the package is initially clean, then during a filling cycle $0 \leq t \leq T$, $m(t) = \delta \int_0^t u(\tau)d\tau$ and $c(t) = \frac{1}{t}\int_0^t u(\tau)d\tau$.

Note that the concentration of agent in the container is the average of input concentration over time. Since we only care about the final concentration in the package, the equation can be simplified as:

$$c(t) = \frac{1}{T}\int_0^t u(\tau)d\tau$$

The transfer function is $H(s) = \frac{1}{Ts}$, which is an integrator.

4.4.3 Risk Assessment via Consequence Simulation

The consequence of a given attack is measured by the number of contaminated packages of finished product in which the concentration of the agent is above a certain threshold. According to the system model in the previous part, the consequence is determined by the mass of inserted agent, the target of attack, and the time of attack (which determines the initial phase of the targeted component). Additionally, since a system with multiple components that is attacked at an upstream component will transfer agent from one component to another, the impact of the attack can also depend on the initial phase of each system component (not only the targeted component) at the time of the attack. Thus, a simulation of agent flow is a straightforward approach to perform a consequence assessment in this setting.

Our consequence assessment tool is implemented in the Simulink toolbox of MATLAB. Given system parameters and attack parameters, the assessment tool provides a measure of the attack consequence. The system parameters include: volume of tanks, flow velocity of liquid, initial phase of each component (if applicable); the attack parameters include: mass of inserted agent, the target of attack and the time of attack, which is equivalent to the initial phase of the targeted component. Assume the attacker has no control over or knowledge of the phasing of the production system; then it may be reasonable to assume that the attack occurs at a random time (initial phase). Under such an assumption, the consequence can be measured by the distribution of the number of contaminated packages or via a summary statistic

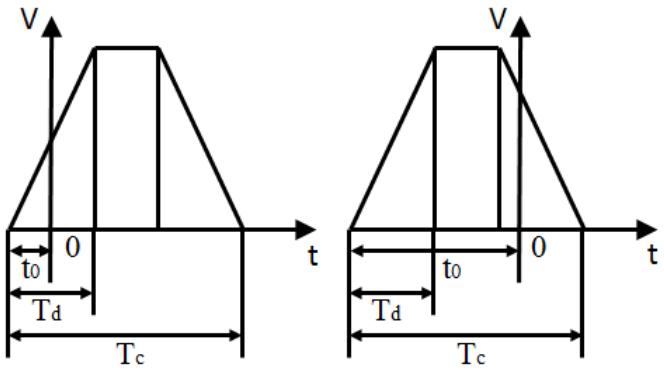

Figure 22: Definition of Initial Phase for a Cyclic System Component

related to the distribution, *i.e.*, average number of contaminated packages.

For a storage tank (or any system component that runs a repeating operating cycle), the initial phase measures the offset of time between the starting time of the operating cycle and the time of attack. Formally, the initial phase of a tank can be represented by time t_0, which is the time elapsed when the agent is inserted, with respect to the starting time of an operating cycle. Alternatively, the initial phase can be represented by a fractional number normalized between zero and one, *i.e.*, $p = t_0/T_c$, where T_c is the duration of an operating cycle of the tank. Figure 22 illustrates the definition of initial phase.

Consider the simplified system in Figure 19 under an attack at a random time. Denote the draining time and cycle time of the raw product tank by T_{d2} and T_{c2}; that of the finished product tank by T_{d3} and T_{c3}. Assume that the attack is on the collecting vat. Suppose at the time of attack, the initial phase of the raw product tank is p_2, and that of the finished product tank is p_3. Note that, $p_2 \in [0, T_{d2}/T_{c2})$ and $p_3 \in [0, (T_{d3} + T_{m3})/T_{c3})$, since the downstream tanks can only receive inflows in these phases. The initial phase of the system is defined by a combination of the initial phases of all components that have phases, *i.e.*, (p_2, p_3) in this example. A random phase of the system means p_2 and p_3 are random in their feasible domains.

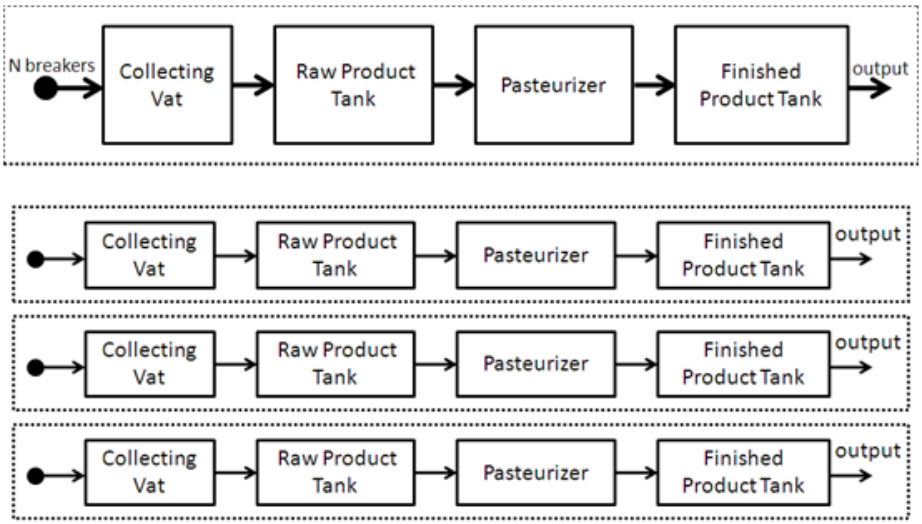

Figure 23: Comparison: Centralized and Decentralized Supply Chain Designs

Table 2: System Parameters of Two Designs

	Centralized	Decentralized	T_m
Collecting Vat	650 L	130 L	N/A
Raw Product Tank	12500 L	2500 L	1800 s
Finished Product Tank	6000 L	1200 L	1200 s
Packaging	10 L	10 L	N/A

In this section, we assume uniform distribution for the phases.

Similarly, if the raw product tank is attacked, the initial phase of the system is represented by (p_2, p_3), where $p_2 \in [0, 1)$ and $p_3 \in [0, (T_{d3} + T_{m3})/T_{c3})$. Note that the attack may occur in the draining phase of the raw product tank. If the finished product tank is attacked, the initial phase of the system is represented by p_3, where $p_3 \in [0, 1)$.

As a proof-of-concept, a case study is developed to illustrate the consequence assessment method. Two different supply chain designs, shown in Figure 23, a centralized and decentralized supply chain, are compared based on their attack consequence. The parameters of both system designs are listed in Table 2.

In Table 2, the second and the third columns list the volume of system component

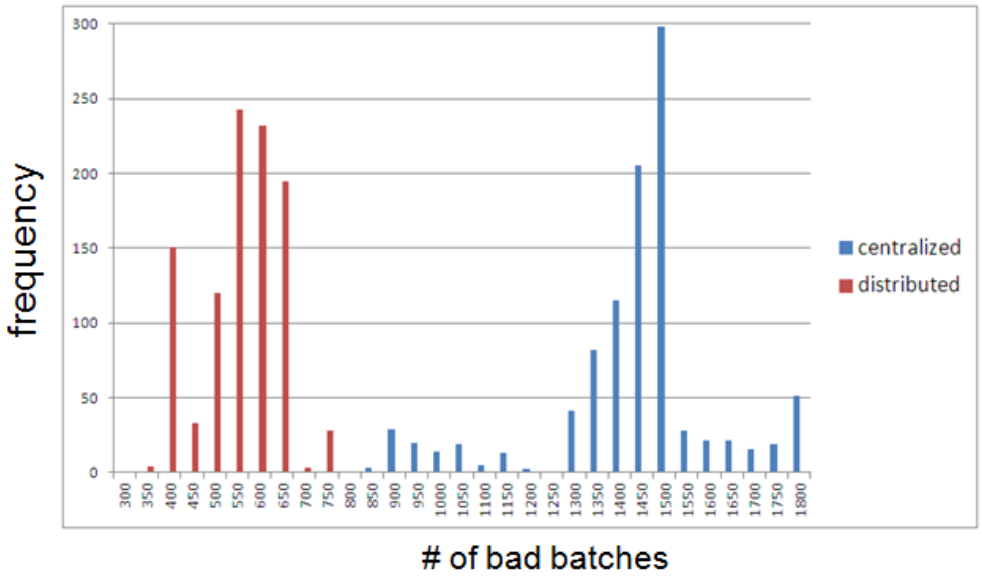

Figure 24: Consequence of Two Supply Chain Designs

tanks and vats in the two systems. The blending phase of the raw product tanks in both systems has the same duration, as does the replenishing phase of the finished product tanks. This phase time data is listed in the column T_m of the table. Suppose the scale of the centralized supply chain is five times the decentralized one. The volumes are measured in liters and the time in seconds. The throughput (or flow velocity) of the two systems are $10L/s$ and $2L/s$ respectively. The pasteurization factor is 0.3 for both systems.

Assume only one target can be attacked in both systems. Suppose the two systems are attacked by the same attack, *i.e.*, a certain type of agent where 1000 grams is inserted to one of the collecting vats of both systems. Then, we compare the consequences of attacking the centralized and decentralized supply chains respectively. After one run of the simulation given a random phase for the system components, the consequence of attacking the centralized supply chain, measured by the number of bad batches, ranges from 819 to 2032, while that of attacking the decentralized one ranges from 350 to 720. The results are presented in a histogram in Figure 24. As we

can see, the consequences caused by attacking the centralized system are significantly higher than the decentralized case. Although it is a common sense that the centralized system has an advantage in productivity due to the economy of scale, the result of the simulation indicates that the centralized system is exposed to a higher risk of an attack. Note that although the throughput is five times higher, the consequences of an attack are not five times higher; the tradeoff between the scale economy of production and this type of scale economy in risk is quite interesting.

4.4.4 Risk Assessment via Worst-case Analysis

The consequence assessment via simulation provides an empirically-computed distribution of consequences, measured in the number of contaminated packages, assuming a random phasing of the system. In this section, we will determine the worst-case consequence by an in-depth analysis on the relationship between the consequence of an attack and the phasing of system. First, we study two special subsystems. Then, we propose an efficient approximation algorithm for worst-case assessment for the whole system, based on the results of the previous parts. Finally, the computational results are presented. Compared to the results obtained by the simulation, our algorithm gives a good estimate of the worst-case consequence.

4.4.4.1 Special Subsystem: Collecting Vat-Raw Product Tank

Consider a simplified system with only two components in sequence: a collecting vat and a raw product tank. The notations are the same as in the previous parts. We will show that the output concentration curve of the raw product tank is a "stair function", regardless of the initial phase of the tank. Figure 25 gives an example of the stair function.

Lemma 4.4.1 *The output curve of the raw product tank is a stair function.*

Proof:

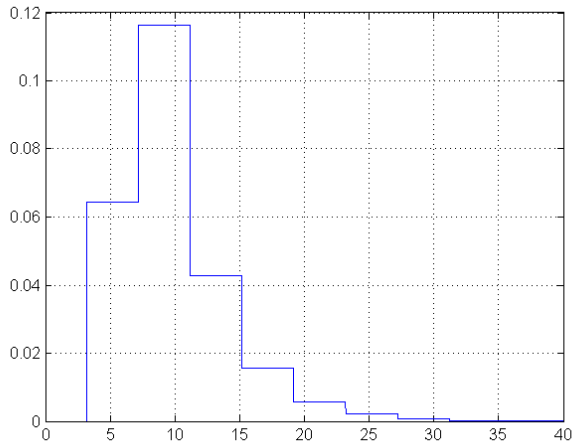

Figure 25: Stair-like function

Suppose the attacker inserts an agent with mass m_0 into the collecting vat, then the output concentration of the collecting vat is $c_1(t) = \frac{m_0}{V_1} e^{-t/T_{d1}}$, where $T_{d1} = V_1/\delta$ is the draining time and V_1 is the volume of collecting vat. Note that the function $c_1(t)$ is monotonically decreasing.

Suppose the initial phase of the raw product tank is $p = t_{02}/T_{c2}$, where t_2 is the initial time of the tank as defined in the previous part. Note that $p \in [0, T_{d2}/T_{c2})$ since the raw product tank only receives inflow during its filling phase. Thus, for the raw product tank, we use $q = pT_{c2}/T_{d2} \in [0, 1)$ to represent its initial phase for convenience. Intuitively, the input concentration curve of the raw product tank, *i.e.*, $c_1(t)$, is segmented by the filling periods, and the output is the average concentration during each filling period. Since the input function is monotonically decreasing, then the output curve is a stair-like function, except for the first "stair". The first "stair" can be higher or lower than the second one, depending on the initial phase.

Note that $t_{02} = qT_{d2}$, where $0 \leq q \leq 1$. The time horizon is segmented into the following half-closed and half-open integral intervals:

$$[0, (1-q)T_{d2}), [(1-q)T_{d2}, (2-q)T_{d2}), \ldots, [(m-1-q)T_{d2}, (m-q)T_{d2}), \ldots$$

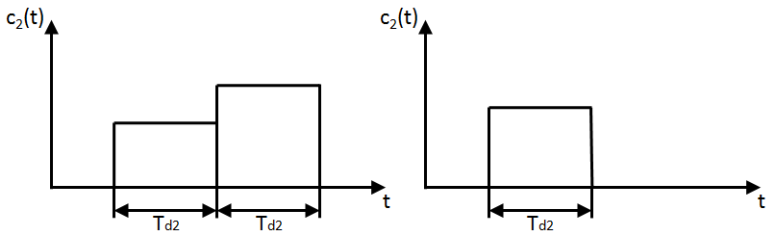

Figure 26: Two Stair Approximation

Suppose the output function is denoted by $c_2(t)$. The segment within the m-th integral interval is represented by $c_{2m}(t)$. Then, the first segment of the output function is a constant level:

$$c_{21}(t) = \frac{1}{T_{d2}} \int_0^{(1-q)T_{d2}} c_1(t)dt = \frac{m_0}{V_1} \frac{T_{d1}}{T_{d2}} [1 - e^{-(1-q)T_{d2}/T_{d1}}]$$

The other segments are also constant levels. For $m \geq 2$, the m-th segment is represented by:

$$c_{2m}(t) = \frac{1}{T_{d2}} \int_{(m-1-q)T_{d2}-t_2}^{(m-q)T_{d2}} c_1(t)dt = \frac{m_0}{V_1} \frac{T_{d1}}{T_{d2}} e^{-(m-q)T_{d2}/T_{d1}} [e^{T_{d2}/T_{d1}} - 1]$$

Hence, the output curve of the raw product tank $c_2(t)$ is a stair function.

Q.E.D.

In practice, however, the "stairs" are seldom observed. The reason is that the raw product tank is usually much larger in volume than the collecting vat. The concentration of the agent always drops below the threshold after at most two filling periods of the raw product tank. Thus, the output function $c_2(t)$ has at most two levels in practice, as illustrated in the left part of Figure 26. In most cases, only one constant level is observed, as illustrated in the right part of Figure 26.

Now we will explore a special case: when the two stairs are of the same height. In this case, the output curve is an interval of width $2T_{d2}$. Suppose a filling period of the raw product tank starts at t^*, as illustrated in Figure 27, such that the first half

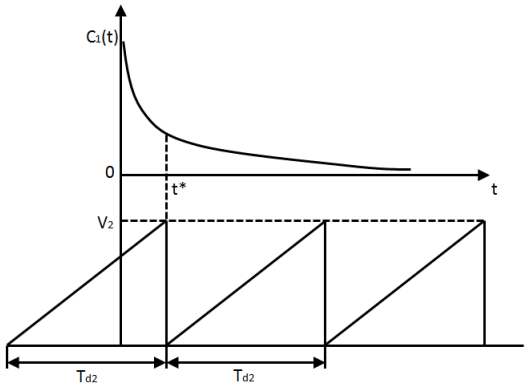

Figure 27: Worst-case Time for Attacking the Collecting Vat

mass of the agent goes into the previous filling cycle. Since the mass of agent flowing into the raw product tank during any time interval $[t_1, t_2]$ is $m = \int_{t_1}^{t_2} \delta c_1(t)dt$, then,

$$\int_0^{t^*} \delta c_1(t)dt = \frac{1}{2}\int_0^{+\infty} \delta c_1(t)dt$$

By solving the equation, we have

$$t^* = T_{d1} \ln 2 \approx 0.693 T_{d1}$$

where t^* is called the "half-life" of the agent in the collecting vat. In addition, the concentration at t^* happens to be half of the initial concentration, *i.e.*, $c_1(t^*) = c(0)/2$.

Given an attack to the collecting vat by inserting an agent of the mass m_0, if the half-life of the agent t^* coincides with the turn-over time of the raw product tank, illustrated in Figure 27, then the output concentration curve is an interval of length $w = 2T_{d2}$. And if the insignificant left-over amount of agent after $t^* + T_d$ is ignored, then the height of the interval can be determined $h \approx \frac{m_0}{2\delta T_{d2}} = m_0/2V_2$.

In this case, by letting $\int_{t^*}^{t^*+T_d} \delta c_1(t)dt \geq \theta$, the minimum mass of the agent can be determined. Then,

$$m_0 \geq 2\theta V_2 (1 - e^{-v_2/V_1})^{-1}$$

If the amount of the agent is only sufficient to generate a single stair above the threshold, then $n^* = T_{d2}/T_s$. We need to have $\frac{1}{T_{d2}} \int_0^{T_{d2}} m_0/V_2 e^{-t/T_{d2}} dt \geq \theta$. Thus,

$$m_0 \geq \theta V_2 (1 - e^{-v_2/V_1})^{-1}$$

If $m_0 < \theta V_2 (1 - e^{-v_2/V_1})^{-1}$, then $n^* = 0$.

In conclusion, the worst-case consequence of the subsystem given an attack at the collecting vat with the mass m_0 is summarized below:

If $m_0 \in [2\theta V_2, +\infty)$, then $n^* = 2T_{d2}/T_s$;

If $m_0 \in [\theta V_2, 2\theta V_2)$, then $n^* = T_{d2}/T_s$;

If $m_0 \in [0, \theta V_2)$, then $n^* = 0$.

Note that if $V_2 >> V_1$, then $e^{-v_2/V_1} \approx 0$.

Finally, we discuss the assumption that $V_2 >> V_1$. For this assumption to hold, a sufficient condition is that the output concentration $c_1(t)$ falls below the threshold within one filling period of the raw product tank. This guarantees that the concentration of the agent drops below the threshold after $t^* + T_d$ in the raw product tank.

Let $c_1(T_{d2}) = \frac{m_0}{V_1} e^{-T_{d2}/T_{d1}} < \theta$. Then, the necessary condition is

$$\frac{V_2}{V_1} > \ln \frac{m_0}{V_1 \theta}$$

Normally, $V_2 > 10V_1$ will be sufficient. In practice, this condition always holds.

4.4.4.2 Special Subsystem: Finished Product Tank with Interval Input

If the agent is inserted into the raw product tank, or if the agent is inserted into the collecting vat when its turn-over time coincides with the half-life of the agent, then both cases reduce to the study of a special subsystem: a finished product tank with an interval input. Suppose the width of the interval is w and the height is h. In this part, we assume $h > \theta$, otherwise, there will be no contaminated batches (in which the concentration of agent is above the threshold).

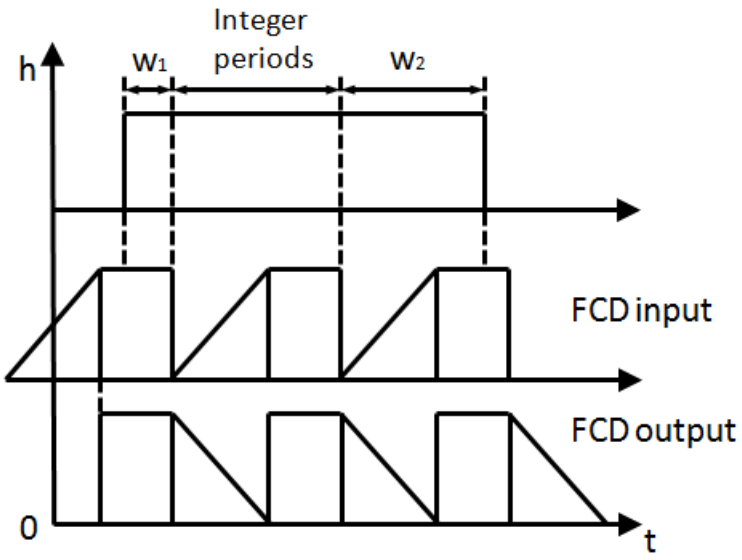

Figure 28: Partial Cycles of the Finished Product Tank

If w is large, then the part of w that is multiple of $(T_{d3} + T_{m3})$ can be easily handled. Assume $h \geq \theta$. A constant input in a full filling period leads to a constant output. The output curve only shifts to the right by T_{d3} in the time horizon. Thus, the number of contaminated batches corresponding to the full filling periods is: $n_{LB} = \frac{T_{d3}+T_{m3}}{T_s} \left\lfloor \frac{w}{T_{d3}+T_{m3}} \right\rfloor$. And an upper bound is $n_{UB} = n_{LB} + \frac{2(T_{d3}+T_{m3})}{T_s}$, since there are at most two partial filling periods left.

For any given w, it can be partitioned into at most three parts: $w = w_1 + w_c + w_2$, illustrated in Figure 28. The full filling periods are the easy part. Without loss of generality, we ignore the full filling periods and assume $0 < w < T_{d3} + T_{m3}$.

Given the partition $w = w_1 + w_2$, w_1 is the part before the full periods and w_2 is the part after them. Since w_1 and w_2 are not determined yet, then we need to establish two functions describing the maximum number of contaminated batches that w_1 and w_2 can produce, respectively.

Suppose the maximum number of contaminated batches that w_1 can produce is $n_1^*(w_1)$, where $0 \leq w_1 < T_d + T_m$. According to the model of the finished product

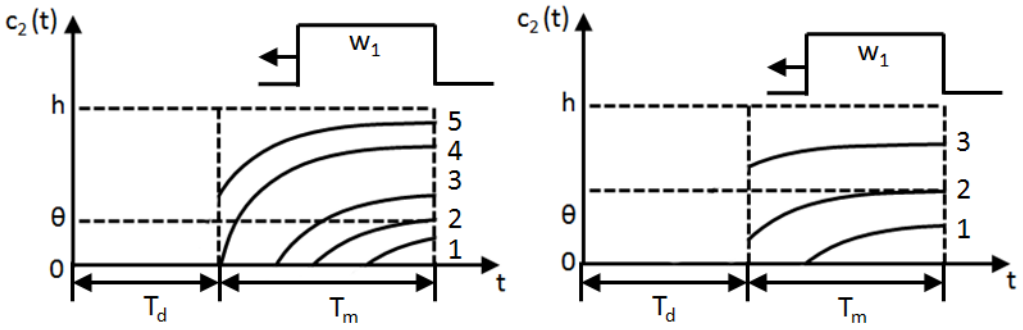

Figure 29: Concentration Curve with respect to w_1

tank, the output corresponding to the replenishing phase T_m is represented as:

$$c(t) = c_0 e^{-t/T_d} + h(1 - e^{-t/T_d})$$

where c_0 is the initial concentration at $t = 0$, the beginning of the replenishing phase. The concentration during the draining phase is the same as the concentration at the end of the replenishing phase. Note that if $w_1 \leq T_m$, then $c_0 = 0$; if $w_1 > T_m$, then $c_0 > 0$.

We compute $n_1^*(w_1)$ as w_1 increases. Figure 29 illustrates the possible concentration curves with different w_1. There are two cases, and we discuss the first case when $T_d \ln \frac{h}{h-\theta} < T_m$, illustrated in the left part of the figure.

(1) If $0 \leq w_1 < T_d \ln \frac{h}{h-\theta} < T_m$, then the whole concentration curve is below the threshold θ, *i.e.*, $c(t) = h(1 - e^{-t/T_d}) < \theta$. Then, $t_1^* = T_d \ln \frac{h}{h-\theta}$ is a critical point obtained by letting $c(t) = \theta$. In this case, $n_1^*(w_1) = 0$.

(2) If $T_d \ln \frac{h}{h-\theta} \leq w_1 < T_m$, then the constant level during the draining phase is always above the threshold. In addition, a part of the curve during the replenishing phase is also above the threshold, lasting for time $w_1 - t_1^*$. Thus, $n_1^*(w_1) = (w_1 + T_d - T_d \ln \frac{h}{h-\theta})/T_s$. Here the worst-case consequence n_1^* is linear to w_1.

(3) If $T_m \leq w_1 < T_m + T_d\theta/h$, then the initial concentration of the replenishing phase $c_0 = \frac{w_1 h}{T_d} > 0$, but below the threshold θ. The time for the curve to reach the threshold is $T_d \ln \frac{h-c_0}{h-\theta}$. Thus, $n_1^*(w_1) = (T_d + T_m - T_d \ln \frac{h-c_0}{h-\theta})/T_s = \frac{1}{T_s}(T_d + T_m -$

$T_d \ln [(\frac{h}{h-\theta})(1 - \frac{w_1 - T_m}{T_d})])$.

The result can be simplified by approximation. Since $\ln(1 + x) \approx x$ when $\mid x \mid$ is close to 0, then $n_1^*(w_1) = \frac{1}{T_s}(T_d + T_m - T_d[\ln(\frac{h}{h-\theta}) + \ln(1 + \frac{T_m - w_1}{T_d})]) = (w_1 + T_d - T_d \ln \frac{h}{h-\theta})/T_s$, which is the same as the previous case. Let $x = \frac{T_m - w_1}{T_d} \in (-\theta/h, 0]$. Since usually h is much larger than θ, the approximation yields a good estimate.

(4) If $T_m + T_d\theta/h \leq w_1 < T_m + T_d$, then the curve is above the threshold from the beginning, *i.e.*, $n_1^*(w_1) = (T_d + T_m)/T_s$.

Consider the other case when $T_d \ln \frac{h}{h-\theta} \geq T_m$, illustrated in the right part of Figure 29. In this case, if starting from zero initial concentration, the concentration curve cannot reach threshold before drained.

This is a degenerate case. In this case, the curve can reach the threshold only with a sufficient c_0. To obtain the critical value for c_0 and then w_1, let $c(T_m) = c_0 e^{-T_m/T_d} + h(1 - e^{-T_m/T_d}) = \theta$, where $c_0 = (w_1 - T_m)h/T_d$.

Let $t_1' = T_m + T_d - T_d e^{T_m/T_d}(h - \theta)/h$. When $w_1 \geq t_1'$, the concentration during draining phase is above the threshold and $n_1^*(w_1) > 0$. The rest of the analysis is similar to the former case.

(1) If $0 \leq w_1 < t_1'$, then $n_1^*(w_1) = 0$.

(2) If $t_1' \leq w_1 < T_m + T_d\theta/h$, then $n_1^*(w_1) = \{T_d + T_m - T_d \ln [(\frac{h}{h-\theta})(1 - \frac{w_1 - T_m}{T_d})]\}/T_s$. This can also be simplified as mentioned earlier. $n_1^*(w_1) = (w_1 + T_d - T_d \ln \frac{h}{h-\theta})/T_s$.

(3) If $T_m + T_d\theta/h \leq w_1 < T_m + T_d$, then $n_1^*(w_1) = (T_d + T_m)/T_s$.

In summary, the maximum number of contaminated batches $n_1^*(w_1)$ can be determined by the following formula. Here we also use the approximation: $\ln(1 + x) = x$.

If $T_d \ln \frac{h}{h-\theta} < T_m$, then let $b_1 = T_d \ln \frac{h}{h-\theta}$; if $T_d \ln \frac{h}{h-\theta} \geq T_m$, then let $b_1 = t_1' = T_m + T_d - T_d e^{T_m/T_d}(h - \theta)/h$.

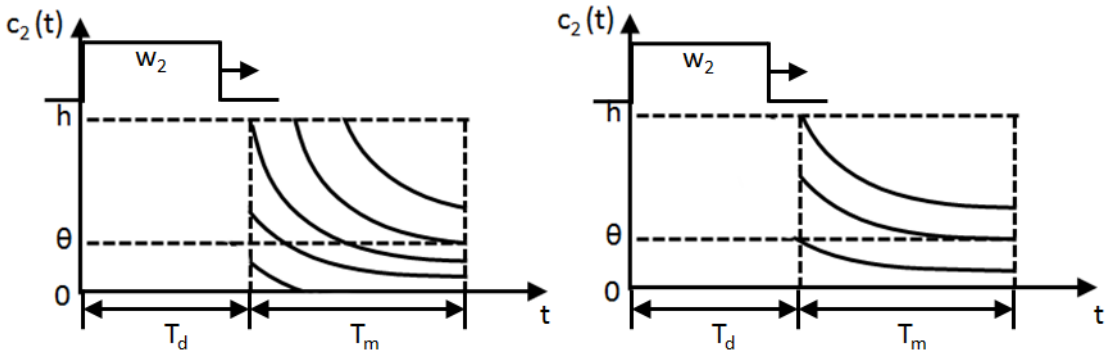

Figure 30: Concentration Curve with respect to w_2

$$n_1^*(w_1) = \begin{cases} 0 & \text{if } 0 \le w_1 < b_1 \\ (w_1 + T_d - T_d \ln \frac{h}{h-\theta})/T_s & \text{if } b_1 \le w_1 < T_m + T_d\theta/h \\ (T_d + T_m)/T_s & \text{if } T_m + T_d\theta/h \le w_1 < T_m + T_d \end{cases}$$

where T_s is the filling time of final package.

Similarly, suppose the maximum number of contaminated batches that w_2 can produce is $n_2^*(w_2)$, where $0 \le w_2 < T_d + T_m$. According to the model of the finished product tank, the output corresponding to the replenishing phase T_m is:

$$c(t) = c_0 e^{-t/T_d}$$

where c_0 is the initial concentration at $t = 0$, the beginning of the replenishing phase. The concentration during the draining phase is the same as the concentration at the end of the replenishing phase. Note that if $w_2 \le T_d$, then $c_0 = w_2 h/T_d < h$; if $w_2 > T_d$, then $c_0 = h$.

We compute $n_2^*(w_2)$ as w_2 increases. Figure 30 illustrates the possible concentration curves with different w_2. There are two cases, and we discuss the first case when $T_d \ln(\frac{h}{\theta}) \le T_m$, illustrated in the left part of the figure.

(1) If $0 \le w_2 < T_d\theta/h$, then $c_0 < \theta$, *i.e.*, the concentration curve is always below the threshold. Thus, $n_2^*(w_2) = 0$.

(2) If $T_d\theta/h \leq w_2 < T_d$, then $c_0 > \theta$ and the concentration curve drops below the threshold before the replenishing phase ends. The critical time t_2^* is obtained by letting $c(t) = c_0 e^{-t/T_d} = \theta$, where $c_0 = w_2 h/T_d$. Then, $t_2^* = T_d \ln \frac{w_2 h}{T_d \theta}$. Thus, $n_2^*(w_2) = t_2^*/T_s = (T_d \ln \frac{w_2 h}{T_d \theta})/T_s$.

(3) If $T_d \leq w_2 < T_d + T_m - T_d \ln \frac{h}{\theta}$, then the concentration curve eventually drops below the threshold before the draining phase starts. Thus, $n_2^*(w_2) = (w_2 - T_d + T_d \ln \frac{h}{\theta})/T_s$.

(4) If $T_d + T_m - T_d \ln \frac{h}{\theta} \leq w_2 < T_d + T_m$, then the concentration curve is always above the threshold. Thus, $n_2^*(w_2) = (T_d + T_m)/T_s$.

Consider the other case when $T_d \ln(\frac{h}{\theta}) > T_m$. This is a degenerate case. Starting with an initial concentration h at $t = 0$, the concentration will not drop below the threshold during the replenishing phase.

In this case, even when $c_0 < h$, $n_2^*(w_2)$ can achieve the maximum value $(T_d + T_m)/T_s$. Let $c(T_m) = c_0 e^{-T_m/T_d} = \theta$, where $c_0 = w_2 h/T_d$. Then, the critical point $t_2' = T_d \frac{\theta}{h} e^{T_m/T_d}$. The rest of the analysis is similar to the former case.

(1) If $0 \leq w_2 < T_d\theta/h$, then $n_2^*(w_2) = 0$.

(2) If $T_d\theta/h \leq w_2 < T_d \frac{\theta}{h} e^{T_m/T_d}$, then, $n_2^*(w_2) = (T_d \ln \frac{w_2 h}{T_d \theta})/T_s$.

(3) If $T_d \frac{\theta}{h} e^{T_m/T_d} \leq w_2 < T_d + T_m$, then, $n_2^*(w_2) = (T_d + T_m)/T_s$.

In summary, the maximum number of contaminated batches $n_2^*(w_2)$ can be determined by the following formula.

Case 1: when $T_d \ln(\frac{h}{\theta}) \leq T_m$,

$$n_2^*(w_2) = \begin{cases} 0 & \text{if } 0 \leq w_2 < T_d\theta/h \\ (T_d \ln \frac{w_2 h}{T_d \theta})/T_s & \text{if } T_d\theta/h \leq w_2 < T_d \\ (w_2 - T_d + T_d \ln \frac{h}{\theta})/T_s & \text{if } T_d \leq w_2 < T_d + T_m - T_d \ln \frac{h}{\theta} \\ (T_d + T_m)/T_s & \text{if } T_d + T_m - T_d \ln \frac{h}{\theta} \leq w_2 < T_d + T_m \end{cases}$$

Case 2: when $T_d \ln(\frac{h}{\theta}) > T_m$,

$$n_2^*(w_2) = \begin{cases} 0 & \text{if } 0 \leq w_2 < T_d\theta/h \\ (T_d \ln \frac{w_2 h}{T_d \theta})/T_s & \text{if } T_d\theta/h \leq w_2 < t_2' \\ (T_d + T_m)/T_s & \text{if } t_2' \leq w_2 < T_d + T_m \end{cases}$$

where $t_2' = T_d \frac{\theta}{h} e^{T_m/T_d}$.

Continue to assume $w = w_1 + w_2 < T_d + T_m$. Given the previous results, the worst-case consequence assessment problem can be reduced to the single-variable optimization problem below:

$$\text{Max } n_1^*(w_1) + n_2^*(w - w_1)$$

$$\text{s.t. } w_1 \in [0, w]$$

We claim that the maximum objective value is achieved at one of the breakpoints of the function $n(w_1) = n_1^*(w_1) + n_2^*(w - w_1)$. But the same maximum value may be achieved at other points. Thus, we only need to compare the objective value at the breakpoints and determining the maximum. To see this, we prove the theorem below:

Theorem 4.4.1 *The maximum value is achieved at one of the breakpoints of the function* $n(w_1) = n_1^*(w_1) + n_2^*(w - w_1)$.

Proof:

Since the functions $n_1^*(x)$ and $n_2^*(x)$ are both piecewise continuous, then the objective function is also piecewise continuous. According to the optimization theory, the maximum can only be achieved at the breakpoints (including the endpoints) or the stationary points, where the derivative with respect to w_1 is zero. Now we show the objective values at the stationary points do not matter. Thus, we only need to check the breakpoints (including the endpoints).

Let $n(w_1) = n_1^*(w_1) + n_2^*(w - w_1)$. Then,

$$\frac{dn(w_1)}{dw_1} = \frac{dn_1^*(w_1)}{dw_1} + \frac{dn_2^*(w - w_1)}{dw_1} = 0$$

Since $w_2 = w - w_1$, then the stationary point is determined by letting

$$\frac{dn_1^*(w_1)}{dw_1} = \frac{dn_2^*(w_2)}{dw_2}\bigg|_{w_2 = w - w_1}$$

Then, we need to determine the derivatives of $n_1^*(w_1)$ and $n_2^*(w_2)$, respectively. Consider computing the derivatives of $n_1^*(w_1)$ first, using the notation in the previous parts.

$$\frac{d}{dw_1} n_1^*(w_1) = \begin{cases} 1/T_s & \text{if } b_1 \le w_1 < T_m \\ 0 & \text{otherwise} \end{cases}$$

where $b_1 = T_d \ln \frac{h}{h-\theta}$, if $T_d \ln \frac{h}{h-\theta} < T_m$; $b_1 = t_1' = T_m + T_d - T_d e^{T_m/T_d}(h - \theta)/h$, if $T_d \ln \frac{h}{h-\theta} \ge T_m$.

Then, compute the derivatives of $n_2^*(w_2)$.

Case 1: when $T_d \ln(\frac{h}{\theta}) \le T_m$,

$$\frac{d}{dw_2} n_2^*(w_2) = \begin{cases} 1/T_s & \text{if } T_d\theta/h \le w_2 < T_d \\ \frac{T_d}{T_s}(w_2)^{-1} & \text{if } T_d \le w_2 < T_d + T_m - T_d \ln \frac{h}{\theta} \\ 0 & \text{otherwise} \end{cases}$$

Case 2: when $T_d \ln(\frac{h}{\theta}) > T_m$,

$$\frac{d}{dw_2} n_2^*(w_2) = \begin{cases} \frac{T_d}{T_s}(w_2)^{-1} & \text{if } T_d\theta/h \le w_2 < t_2' \\ 0 & \text{otherwise} \end{cases}$$

where $t_2' = T_d \frac{\theta}{h} e^{T_m/T_d}$.

First, we will ignore the case when $\frac{d}{dw_1} n_1^*(w_1) = 0$ or $\frac{d}{dw_2} n_2^*(w_2) = 0$. In this case, the derivative of the objective function $\frac{d}{dw_1} n(w_1) = 0$ only if both of the above derivatives are equal to zero. This implies a constant objective value $n(w_1)$ in this

range. Thus, the objective value at the stationary point is equal to that at some breakpoint.

Then, $\frac{d}{dw_1}n_1^*(w_1) = 1/T_s$ in an interval. We will ignore the case when $\frac{d}{dw_2}n_2^*(w_2)$ is the same constant, since this also implies a constant objective value $n(w_1)$. And the objective value at the stationary point is equal to that at some breakpoint.

The only other possibility that $n'(w_1) = 0$ is the case that $\frac{T_d}{T_s}(w_2)^{-1} = 1/T_s$, which implies $w_2 = T_d$. In this case, the stationary point coincides with the breakpoint $w_2 = T_d$.

Hence, the stationary points of the function $n(w_1)$ do not matter. Thus, the maximum value is achieved at one of the breakpoints of the function $n(w_1) = n_1^*(w_1) + n_2^*(w - w_1)$.

Q.E.D.

According to Theorem 4.4.1, the optimization problem reduces to a easy problem, *i.e.*, comparing the function value at a few breakpoints and determining the maximum. Suppose the breakpoints of $n_1^*(x)$ is denoted by the set S_1 and those of $n_2^*(x)$ is denoted by the set S_2. Then the breakpoints of the objective function are represented by the set: $S = \{w_1 \in [0, w] : w_1 \in S_1 \text{ or } w - w_1 \in S_2\}$. It is easy to determine the maximum on the breakpoints by comparison.

One technical issue during the implementation needs to be mentioned here. Since the worst-case function is not continuous, then, the value in the neighborhood of a breakpoint can change dramatically. A better way to compute the objective value at a breakpoint is provided here. Suppose b is a breakpoint of the function $f(x)$, then the function value is computed by: $f(x) = \max\{f(b - \epsilon), f(b + \epsilon)\}$, where ϵ is a sufficiently small number.

However, there is a remaining case we need to discuss in order to complete the worst-case analysis on finished product tank with interval input. If $w < T_m + T_d$,

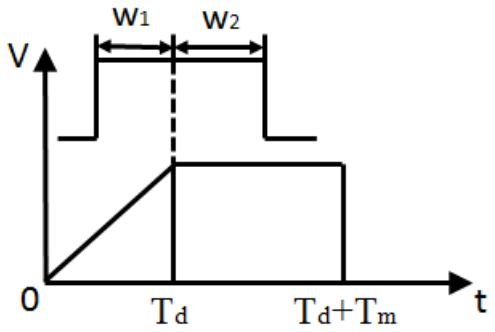

Figure 31: Input Interval in One Period

then the input interval of width w may be contained in one filling period. Without loss of generality, we assume w can be partitioned into two parts w_1 and w_2. w_1 is in the filling phase and w_2 is in the replenishing phase, as illustrated in Figure 31.

Suppose c_0 is the initial concentration at the beginning of the replenishing phase; c_r is the concentration when the input drops to zero; c_e is the concentration at the end of the replenishing phase.

$c_0 = w_1 h/T_d$.

$c_r = h - (h - c_0)e^{-w_2}/T_d = h - (h - w_1 h/T_d)e^{-w_2}/T_d$

$c_e = c_r e^{-(T_m - w_2)/T_d} = he^{(w_2 - T_m)/T_d} - (h - w_1 h/T_d)e^{-T_m/T_d}$

Since the raw product tank is usually larger than the finished product tank, then it is usual that $w > T_m + T_d$. Thus, this case is rare. If necessary for this case, we can use a linear search over $w_1 \in [0, w]$ to obtain the maximum number of contaminated batches.

4.4.4.3 Complete System and Computational Results

Consider the four-component liquid egg production system in Figure 19. It is computationally time-consuming to obtain the exact worst-case consequence of the whole system. However, the analysis can be substantially simplified under an assumption. And we will show by computational experiments that the simplified method efficiently yields a fairly good approximation of the worst-case consequence.

In the previous two parts, we propose efficient methods to analyze two special subsystems. Recall that the worst-case output of the raw product tank is a two-level interval function. If the two levels are of the same height, then the worst-case consequence of the whole system is easy to assess by integrating the results of the two subsystems. Note that this assumption may lead to a suboptimal solution. But since the mass of the agent used for attack is usually sufficient, then the concentration level out of the raw product tank is usually high compared to the threshold, and thus, even the two different levels makes no difference. And the experiments show that the simplified method gives a good approximation of the worst-case consequence in practice.

Below is a summary of the worst-case assessment method, based on the assumption above. Suppose the agent of mass m_0 is inserted into the collecting vat. If $m_0 \in [2\theta V_2, +\infty)$, then the raw product tank outputs an interval of width $w = 2T_{d2}$ and height $h = m_0/(2T_{d2})$; if $m_0 \in [\theta V_2, 2\theta V_2)$, then the output interval is $w = T_{d2}$ and height $h = m_0/T_{d2}$; if $m_0 \in [0, \theta V_2)$, then the attack produces no contaminated batches, *i.e.*, $n^* = 0$.

Given the interval of width w and height h as the input to the finished product tank, the worst-case consequence is determined according to the method provided in the previous part. Note that the worst-case consequence obtained is an approximation under the assumption above.

Note that if the height h of the output of the raw product tank is close to the threshold θ, the assumption may lead to a bad estimation of the worst-case. On the contrary, if the mass is sufficient large, then $h >> \theta$. In this case, the assumption holds.

Consider the case study comparing two supply chain designs. The same setting is used as in the simulation. Suppose $m_0 = 1000$ grams of the agent is inserted into one of the collecting vats of both systems.

Consider the centralized supply chain. Since $V_2 > 10V_1$ and $m_0 > 2\theta V_2$, then the output of the raw product tank is an interval of width $w = 2T_{d2} = 2500$ sec and height $h = m_0/(2V_2) = 0.04g/L$. After going through the pasteurizer, the height is adjusted to $h' = \alpha h = 0.012g/L$. For the convenience of notation, h is used to replace h' in the next paragraphs.

Since $w > T_{d3}+T_{m3}$, then $n_{LB} = 1800$ and w is adjusted to be $w = w_1+w_2 = 700$. Consider $n_1^*(w_1)$. The function is not degenerate because $T_{d3} \ln h/(h-\theta) < T_m$. There are only two breakpoints for $w_1 \in [0, 700]$. For both breakpoints $w_1 = 0$ and $w_1 = 700$, the function $n_1^*(w_1) = 0$. Consider $n_2^*(w_2)$. The function is not degenerate because $T_{d3} \ln h/\theta < T_m$. We also consider the breakpoints of w_2, which correspond to $w_1 \in [0, 700]$. At the breakpoints, $w_2 = 500$, $n_2^*(500) = 0$; $w_2 = 600$, $n_2^*(600) = 109$. Since the derivative function of $n_1^*(w_1)$ is constant zero for $w_1 \in [0, 700]$, then we do not need to check the stationary points, since they are coincide with the breakpoints if they exist. The left part of Table 3 lists the breakpoints of both cases.

Comparing the values of $n_1^*(w_1)+n_2^*(w-w_1)$ at breakpoints $w_1 \in \{0, 100, 200, 700\}$, we find the maximum value is achieved $n^* = 209$ at $w_1 = 0$. Thus, the worst-case consequence for the centralized supply chain is 2009, after adding n_{LB}.

The maximum value that the simulation provides is 2032. Thus, the worst-case assessment method provides a good estimate, with a gap of 1.13%. Note that the simulation may not provide the true worst-case, since it is based on random phasing, not enumeration.

Similarly, consider the decentralized supply chain. Since $V_2 > 10V_1$ and $m_0 > 2\theta V_2$, then the output of the raw product tank is an interval of width $w = 2T_{d2} = 2500$sec and height $h = m_0/(2V_2) = 0.2g/L$. After going through the pasteurizer, the height is adjusted to $h' = \alpha h = 0.06g/L$.

Since $w > T_{d3} + T_{m3}$, then $n_{LB} = 360$ and w is adjusted to be $w = w_1 + w_2 = 700$. Consider $n_1^*(w_1)$. The function is not degenerate. There are only two breakpoints

Table 3: Breakpoints of the Worst-case Function

Centralized supply chain					Decentralized supply chain				
w_1	w_2	n_1^*	n_2^*	n^*	w_1	w_2	n_1^*	n_2^*	n^*
0	700	0	209	209	0	700	0	235	235
100	600	0	109	109	100	600	0	215	215
200	500	0	0	0	109	591	120	213	333
700	0	0	0	0	600	100	218	0	218
					700	0	238	0	238

for $w_1 \in [0, 700]$. For $w_1 = 0$, $n_1^*(w_1) = 0$; For $w_1 = 109$, $n_1^*(w_1) = 120$. Consider $n_2^*(w_2)$. The function is also not degenerate. Also consider the breakpoints of w_2, which correspond to $w_1 \in [0, 700]$. At the breakpoints, $w_2 = 0$, $n_2^*(0) = 0$; $w_2 = 600$, $n_2^*(600) = 219$. After checking, there is no stationary points for n^*. Comparing the values of $n_1^*(w_1) + n_2^*(w - w_1)$ at breakpoints $w_1 \in \{0, 109, 100, 700\}$, we find the maximum value is achieved $n^* = 120 + 213 = 333$ at $w_1 = 109$. Thus, the worst-case consequence for the centralized supply chain is 693, after adding n_{LB}. The right part of Table 3 lists the breakpoints of both cases.

The maximum value that the simulation provides is 720. Thus, the worst-case assessment method provides a good estimation, with a gap of 3.75%.

4.5 Summary and Future Research

In this chapter, we present a methodology for performing risk assessment on food supply chains, which may be intentionally used for chemical or biological attacks. First, a procedure is provided to assess this type of risk for a general food supply chain. However, the challenging part is to quantify the risk and then assess the risk in a reasonable model of a general food supply chain. After capturing the core of the problem, we propose a general modeling scheme on the basis of state-space models, which can be used for a variety of follow-up assessment and optimization.

To give a concrete illustration of the methodology, a case study on the liquid egg

supply chain is presented. Based on the industry data, a system model of the liquid egg supply chain is built under several reasonable simplifying assumptions. Then, the consequence assessment is performed by running a simulation assuming a random attack to the system. To improve the efficiency of the assessment, an in-depth analysis is carried out to determine the worst-case consequence given a smart attack, which takes into account the operational characteristics of the supply chain. Finally, the results from the simulation and the worst-case analysis are compared. Fast evaluation of consequence is critical if the consequence assessment tool will support the design or optimization of the system.

Our risk assessment tool can provide quantitative support for further analyses. For example, it can be embedded into a decision support system for protecting the system. A gaming model can be built based on the assessment tool, to determine the allocation of the defensive resources. In the future, our research will focus on applying the risk assessment method to improve the design and control of the food supply chains. The decision-making is based on a trade-off between the vulnerability and the productivity of the supply chain.

APPENDIX A

BASICS OF LAPLACE TRANSFORM

In mathematics, the Laplace transform is a widely used integral transform. The Laplace transform has many important applications throughout the sciences, especially for solving differential equations.

Suppose a function $f(t)$ has a real argument $t(t \geq 0)$. Laplace transform maps the real function to a function $F(s)$ with a complex argument s. This transformation is essentially bijective for the majority of practical uses; the respective pairs of $f(t)$ and $F(s)$ are matched in tables. The Laplace transform has the useful property that many relationships and operations over the originals $f(t)$ correspond to simpler relationships and operations over the images $F(s)$, so sometimes it is easier to analyze the image $F(s)$ instead of $f(t)$.

A.1 Definition

The Laplace transform of a function $f(t)$, defined for all real numbers $t \geq 0$, is the function $F(s)$, defined by

$$F(s) = \int_0^\infty f(t)e^{-st}dt$$

where $s = \delta + i\omega$ is a complex number. The Laplace transform is also written as $\mathcal{L}\{f(t)\}$.

Given $F(s)$, $f(t)$ can be obtained by the inverse Laplace transform, which is defined by the following complex integral:

$$f(t) = \mathcal{L}^{-1}\{f(t)\} = \frac{1}{2\pi i}\lim_{T\to\infty}\int_{\gamma+iT}^{\gamma+iT} F(s)e^{st}ds$$

where γ is a real number so that the contour path of integration is in the region of convergence of $F(s)$.

Table 4: Properties of Laplace Transform

	Time domain	s domain
Linearity	$af(t) + bg(t)$	$aF(s) + bG(s)$
Differentiation	$f'(t)$	$sF(s) - f(0)$
Integration	$\int_0^t f(\tau)d\tau$	$\frac{1}{s}F(s)$
Time shifting	$f(t-\alpha)u(t-\alpha)$	$F(s)e^{-\alpha s}$

Table 5: Laplace transform of common functions

	$f(t)$	$\mathcal{L}\{f(t)\}$
unit impulse	$\delta(t)$	1
unit step	$u(t)$	$\frac{1}{s}$
exponential decay	$e^{-\alpha t}u(t)$	$\frac{1}{s+\alpha}$
exponential approach	$(1-e^{-\alpha t})u(t)$	$\frac{\alpha}{s(s+\alpha)}$
sine	$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
cosine	$\cos \omega t$	$\frac{s}{s^2+\omega^2}$

A.2 Properties

The Laplace transform has a number of properties that make it useful for analyzing linear dynamical systems. The most significant advantage is that differentiation and integration become multiplication and division, respectively, by s. In practice, the definitions above are seldom used to obtain the Laplace transform and inverse Laplace transform. Instead, they are often obtained by the Laplace transform of common functions, using good properties of the transform. We list several properties often used in Table 4. Denote $F(s) = \mathcal{L}\{f(t)\}$ and $G(s) = \mathcal{L}\{g(t)\}$.

A.3 Laplace transform of common functions

The Laplace transform of common functions are listed in Table 5.

REFERENCES

[1] Ahuja, R., Magnanti, T., and Orlin, J., *Network flows: theory, algorithms, and applications*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[2] Ali, A., Allen, E., Barr, R., and Kennington, J., "Reoptimization procedures for bounded variable primal simplex network algorithms," *European journal of operational research*, vol. 23, pp. 256–263, 1986.

[3] Archetti, C., Speranza, M., and Savelsbergh, M., "An optimization-based heuristic for the split delivery vehicle routing problem," *Transportation Science*, vol. 42, no. 1, pp. 22–31, 2008.

[4] Baker, "A Vulnerability Assessment Methodology for Critical Infrastructure Facilities," critical infrastructure protection program workshop proceedings, George Mason University Press, 2004.

[5] Balakrishnan, A., Mirchandani, P., and Natarajan, H. P., "Connectivity upgrade models for survivable network design," *Operations Research*, vol. 57, pp. 170–186, 2009.

[6] Bertsimas, D. and Tsitsiklis, J., *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, 1997.

[7] Bier, V. M., "Balancing terrorism and natural disasters–defensive strategy with endogenous attacker effort," *Operations Research*, vol. 55, pp. 976–991, 2007.

[8] Bier, V., Nagaraj, A., and Abhichandani, V., "Protection of simple series and parallel systems with components of different values," *Reliability Engineering System Safety*, vol. 87, pp. 315–323, 2005.

[9] Bigun, E., "Risk analysis of catastrophes using experts' judgments: an empirical study on risk analysis of major civil aircraft accidents in europe," *European Journal of Operational Research*, vol. 87, p. 599C612, 1995.

[10] C. L. Monma, D. F. S., "Methods for designing communications networks with certain two-connected survivability constraints," *Operations Research*, vol. 37, pp. 531–541, 1989.

[11] Chopra, S. and Sodhi, M., "Managing risk to avoid supply chain breakdown," *MIT Sloan Management Review*, vol. 46, pp. 53–61, 2004.

[12] Christopher, M. and Peck, H., "Building the resilient supply chain," *International Journal of Logistics Management, The*, vol. 15, no. 2, pp. 1–14, 2004.

[13] Crainic, T., "Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, pp. 272–288, 2000.

[14] Crainic, T. and Gendreau, M., "Cooperative parallel tabu search for capacitated network design," *Journal of Heuristics*, vol. 8, no. 6, pp. 601–627, 2002.

[15] Crainic, T., Gendreau, M., and Farvolden, J., "A simplex-based tabu search method for capacitated network design," *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 223–236, 2000.

[16] Dalziel, G. and Ng, S., "Security, safety and defence: food for thought." 2009.

[17] Dantzig, G. B., "Application of the simplex method to a transportation problem," *Activity Analysis of Production and Allocations*, pp. 359–373, 1951.

[18] Dantzig, G., *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press, 1963.

[19] Erera, A., Hewitt, M., Savelsbergh, M., and Zhang, Y., "Improved load plan design through integer programming based local search," 2009. submitted for publication.

[20] Farvolden, J. and Powell, W., "Subgradient methods for the service network design problem," *Transportation Science*, vol. 28, no. 3, pp. 256–272, 1994.

[21] Franceschi, R., Fischetti, M., and Toth, P., "A new ilp-based refinement heuristic for vehicle routing problems," *Mathematical Programming*, vol. 105, no. 2, pp. 471–499, 2006.

[22] Gardner, J. and Cooper, M., "Strategic supply chain mapping approaches," *Journal of Business Logistics*, vol. 24, pp. 37–64, 2003.

[23] Ghamlouch, I., Crainic, T., and Gendreau, M., "Cycle-based neighborhoods for fixed charge capacitated multicommodity network design," *Operations Research*, vol. 51, pp. 655–667, 2003.

[24] Ghamlouch, I., Crainic, T., and Gendreau, M., "Path relinking, cycle-based neighborhoods and capacitated multicommodity network design," *Annals of Operations Research*, vol. 131, pp. 109–133, 2004.

[25] Goemans, M. X. and Bertsimas, D. J., "Survivable networks, linear programming relaxations and the parsimonious property," *Mathematical Programming*, vol. 60, pp. 145–166, 1993.

[26] Goldberg, S., Davis, S., and Pegalis, A., *Y2K Risk Management: Contingency planning, business continuity, and avoiding litigation*. John Wiley & Sons, Inc., 1999.

[27] Greenhouse, S., "Both Sides See Gains in Deal to End Port Labor Dispute," *New York Times*, vol. Nov 25, p. A14, 2002.

[28] Grotschel, M., Monma, C., and Stoer, M., "Design of survivable networks," *Handbooks in Operations Research and Management Science*, vol. 7, pp. 617–672, 1995.

[29] Hamberg, W. A., "TRANSPORTATION VULNERABILITY RESEARCH: REVIEW AND APPRAISAL 1959-1969," Report AD0699423, Stanford Research Institute, Menlo Park, CA, Janurary 1969.

[30] Helferich, O. and Cook, R., *Securing the Supply Chain*. Oak Brook, IL: Council of Supply Chain Management Professionals, 2002.

[31] Jarrah, A., Johnson, E., and Neubert, L., "Large-scale, less-than-truckload service network design," *Operations research*, vol. 57, no. 3, pp. 609–625, 2009.

[32] Johnson, K. and Kort, J., "2004 Redefinition of the BEA Economic Areas," *Survery of Current Business*, pp. 68–75, November 2004.

[33] Jüttner, U., Peck, H., and Christopher, M., "Supply chain risk management: outlining an agenda for future research," *International Journal of Logistics: Research and Applications*, vol. 6, no. 4, pp. 197–210, 2003.

[34] Kaplan, S. and Garrick, B., "On the quantitative definition of risk," *Risk analysis*, vol. 1, no. 1, pp. 11–27, 1981.

[35] Lindsey, K., *Improvements in freight consolidation networks through IP-based local search*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2012.

[36] Major, J., "Advanced techniques for modeling terrorism risk," *White Paper*, 2002.

[37] Manning, L., Baines, R., and Chadd, S., "Deliberate contamination of the food supply chain," *British Food Journal*, vol. 107, no. 4, pp. 225–245, 2005.

[38] Michael Stamatelatos et al, "Probabilistic Risk Assessment Procedure Guide for NASA Managers and Practitioners," guide, NASA Headquarters, Washington, DC, August 2002.

[39] Miller, K., "A framework for integrated risk management in international business," *Journal of international business studies*, pp. 311–331, 1992.

[40] Mitroff, I. and Alpaslan, M., "Preparing for evil," *Harvard Business Review*, vol. 81, p. 109C115, 2003.

[41] Narasimhan, R. and Talluri, S., "Perspectives on risk management in supply chains," *Journal of Operations Management*, vol. 27, pp. 114–118, 2009.

[42] O'Ryan, M., Djuretic, T., Wall, P., Nichols, G., Hennessy, T., Slutsker, L., Hedberg, C., MacDonald, K., and Osterholm, M., "An outbreak of salmonella infection from ice cream," *New England Journal of Medicine*, vol. 335, no. 11, pp. 824–825, 1996.

[43] Peck, H., "Drivers of supply chain vulnerability : an integrated framework," *International Journal of Physical Distribution and Logistics Management*, vol. 35, pp. 210–232, 2005.

[44] Powell, R., "Defending against terrorist attacks with limited resources," *American Political Science Review*, vol. 101, pp. 527–541, 2007.

[45] Powell, W., "A local improvement heuristic for the design of less-than-truckload motor carrier networks," *Transportation Science*, vol. 20, no. 4, pp. 246–257, 1986.

[46] Powell, W. and Koskosidis, I., "Shipment routing algorithms with tree constraints," *Transportation Science*, vol. 26, no. 3, pp. 230–245, 1992.

[47] Powell, W. and Sheffi, Y., "The load planning problem of motor carriers: Problem description and a proposed solution approach," *Transportation Research Part A: General*, vol. 17, no. 6, pp. 471–480, 1983.

[48] Powell, W. and Sheffi, Y., "Design and implementation of an interactive optimization system for network design in the motor carrier industry," *Operations Research*, pp. 12–29, 1989.

[49] Raghavan, S. and Magnanti, T., "Network connectivity," *Annotated Bibliographies in Combinatorial Optimization*, pp. 335–354, 1997.

[50] Rao, S. and Goldsby, T., "Supply chain risks: a review and typology," *International Journal of Logistics Management, The*, vol. 20, no. 1, pp. 97–123, 2009.

[51] Sampson, A. and Smith, R., "Assessing risk through the determination of rare event," *Operations Research*, vol. 30, p. 839C866, 1982.

[52] Savelsbergh, M. and Song, J., "An optimization algorithm for the inventory routing problem with continuous moves," *Computers & Operations Research*, vol. 35, no. 7, pp. 2266–2282, 2008.

[53] Schmid, V., Doerner, K., Hartl, R., Savelsbergh, M., and Stoecher, W., "A hybrid solution approach for ready-mixed concrete delivery," *Transportation Science*, vol. 43, no. 1, pp. 70–85, 2009.

[54] Science Applications International Corporation (SAIC), "A Guide to Highway Vulnerability Assessment for Critical Asset Identification and Protection," contractor's report, Transportation Policy and Analysis Center, VA, May 2002.

[55] Sheffi, Y., *The Resilient Enterprise*. Massachusetts Institute of Technology, Cambridge, MA: The MIT Press, 2005.

[56] Sheffi, Y., "The resilient enterprise: overcoming vulnerability for competitive advantage," *MIT Press Books*, vol. 1, 2007.

[57] Sobel, J., Khan, A., and Swerdlow, D., "Threat of a biological terrorist attack on the us food supply: the cdc perspective," *The Lancet*, vol. 359, no. 9309, pp. 874–880, 2002.

[58] Srinivasan, V. and Thompson, G., "An operator theory of parametric programming for the transportation problem–part one," *Naval research logistics Quarterly*, vol. 19, pp. 205–225, 1972.

[59] Srinivasan, V. and Thompson, G., "An operator theory of parametric programming for the transportation problem–part two," *Naval research logistics Quarterly*, vol. 19, pp. 227–252, 1972.

[60] Steiglitz, K., Weiner, P., and Kleitman, D., "The design of minimum-cost survivable networks," *IEEE Transactions on Circuit Theory*, vol. 16, pp. 455–460, 1969.

[61] Surface Transportation Board, "Public Use Waybill." `http://www.stb.dot.gov/stb/industry/econ_waybill.html/`, 2007. [Online; accessed 19-November-2009].

[62] Svensson, G., "Key areas, causes and contingency planning of corporate vulnerability in supply chains," *International Journal of Physical Distribution and Logistics Management*, vol. 34, pp. 728–748, 2004.

[63] United States Department of Agriculture, "Grain transportation report," *USDA Report*, 2007.

[64] U.S. Department of Energy, "Vulnerability Assessment Methodology–Electric Power Infrastructure," draft, NASA Headquarters, Washington, DC, September 2002.

[65] Wein, L. and Liu, Y., "Analyzing a bioterror attack on the food supply: the case of botulinum toxin in milk," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 28, p. 9984, 2005.

[66] White, J., "Katrina Cuts LA. Grain Exports," *Money*, 2005.

[67] Wieberneit, N., "Service network design for freight transportation: a review," *OR spectrum*, vol. 30, no. 1, pp. 77–112, 2008.

[68] Wikipedia website, "Hurricane Katrina." `http://en.wikipedia.org/wiki/Hurricane_Katrina/`, 2012. [Online; accessed 20-November-2012].

[69] Wood, N., Good, J., and Goodwin, R., "Vulnerability assessment of a port and harbor community to earthquake and tsunami hazards: Integrating technical expert and stakeholder input," *Natural Hazards Review*, vol. 3, no. 4, pp. 148–157, 2002.

[70] Wu, T., Blackhurst, J., and OGrady, P., "Methodology for supply chain disruption analysis," *International Journal of Production Research*, vol. 45, p. 1665C1682, 2007.